Prozessunterstützung für den Entwurf von Wearable-Computing-Systemen

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades Dr. rer. nat.

vorgelegt von
Dipl.-Inform. Tobias Klug
geboren in Frankfurt

Tag der Einreichung: 15.4.2008
Tag der Disputation: 30.5.2008

Referenten: Prof. Dr. Max Mühlhäuser, Darmstadt
Prof. Dr.-Ing. Ralph Bruder, Darmstadt

Darmstadt 2008

Darmstädter Dissertationen
D17

Danksagung

Diese Dissertation ist weder Steffi Graf, noch André Agassi gewidmet. Dafür jedoch den vielen Personen, die mich in den letzten dreieinhalb Jahren begleitet und unterstützt haben.

An erster Stelle möchte ich hier meinem Doktorvater Prof. Max Mühlhäuser für die Anregungen und Diskussionen danken, die entscheidend zum Entstehen dieser Arbeit beigetragen haben.

Außerdem möchte ich meiner Freundin Nina Steinert und meinen Eltern danken, die meine Entscheidung zu promovieren unterstützt und mich die gesamte Zeit über begleitet haben. Sie haben mich immer wieder motiviert und auch bei der Korrektur des endgültigen Textes tatkräftige Unterstützung geleistet.

Des weiteren möchte ich Markus Roth, Hristo Indzhov und Svenja Kahn danken, die mit ihren Diplom- und Bachelorarbeiten entscheidend zur Entstehung dieser Arbeit beigetragen haben.

Außerdem möchte ich den vielen Kollegen an der Universität und bei SAP Research für die freundliche und lockere Arbeitsatmosphäre danken. Victoria Carlsson, Andreas Zinnen und Thomas Ziegert danke ich für die gute Zusammenarbeit und fruchtbaren Diskussionen im Rahmen des wearIT@work-Projektes und auch darüber hinaus. Ich danke meinen Zimmerkollegen an der Universität Melanie Hartmann und Fernando Lyardet, die sich immer wieder meine wilden Ideen anhören und kommentieren mussten. Für die gute Zusammenarbeit danke ich weiterhin Daniel Schreiber der zum Glück einfach nicht nein sagen kann, wenn man ihn um Hilfe bittet.

Andreas Heinemann und Dirk schnell möchte ich zusätzlich für die vielen “technischen” Hinweise zur Erstellung dieser Arbeit danken.

Viele weitere Helfern und Diskussionspartnern sind ungenannt geblieben. Auch ihnen möchte ich an dieser Stelle danken.

Ehrenwörtliche Erklärung [1]

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades "Dr. rer. nat." mit dem Titel "Prozessunterstützung für den Entwurf von Wearable-Computing-Systemen" selbständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Darmstadt, den 15.4.2008

Dipl.-Inform. Tobias Klug

[1] Gemäß §9 Abs. 1 der Promotionsordnung der TU Darmstadt

Wissenschaftlicher Werdegang des Verfassers[2]

1985–1989	Käthe-Kollwitz Grundschule, Frankfurt
1989–1995	Leibniz Gymnasium, Frankfurt
1995–1998	Abitur, Friedrich Dessauer Gymnasium, Frankfurt
1999–2004	Studium der Informatik an der Technischen Universität Darmstadt
2001–2003	Studentische Hilfskraft am Lehrstuhl Graphisch-Interaktive Systeme
2003–2004	Diplomarbeit am Lehrstuhl Graphisch-Interaktive Systeme Technische Universität Darmstadt “Meshfree Radiosity”
2004–2007	Wissenschaftlicher Mitarbeiter an der Technischen Universität Darmstadt
seit 2004	Wissenschaftlicher Mitarbeiter im SAP Research CEC Darmstadt

[2]Gemäß §20 Abs. 3 der Promotionsordnung der TU Darmstadt

Zusammenfassung

Motivation.
In Beruf und Alltag führen Menschen häufig Tätigkeiten aus, die mit der gleichzeitigen Bedienung eines herkömmlichen Computersystems nicht vereinbar sind. Fortschritte in der Hardware-Entwicklung ermöglichen aber den Entwurf maßgeschneiderter Computersysteme für eine wachsende Zahl von Arbeitsumgebungen. Damit können Medienbrüche verhindert und Prozesse optimiert werden, ohne die „Primäraufgabe“, d.h. die eigentliche Tätigkeit des Benutzers, zu behindern.

Forschungsgegenstand: Wearable Computing.
Wearable Computing bezeichnet das Forschungsgebiet, welches sich mit der Gestaltung von Hardware-Software-Systemen für Arbeitssituationen widmet. Hier werden Geräte eingesetzt, die wie ein Kleidungsstück permanent ihre Funktion erfüllen, den Träger jedoch möglichst weder behindern noch stören. Viele dieser Geräte trägt der Benutzer tatsächlich direkt am Körper, wodurch sie jederzeit zur Verfügung stehen. Hinzu kommen Geräte, die vom Benutzer für spezielle Teilaufgaben als Werkzeuge aufgegriffen werden.

Wissenschaftliche Fragestellung und Ziele.
Während die technischen Grundlagen für Wearable-Computing-Anwendungen weit fortgeschritten sind, werden frühere Phasen des Entwurfs nicht ausreichend unterstützt. Die Verzahnung von Primäraufgabe und Wearable-Computing-System macht die intensive Einbindung des späteren Benutzers in den Entwurfsprozess notwendig. Bestehende *benutzerorientierte Entwurfsprozesse* bieten jedoch derzeit noch keine Unterstützung für die *Berücksichtigung von spezifischen Aspekten* des Wearable Computing. Ziel dieser Arbeit ist es daher, den Entwurfsprozess für Wearable-Computing-Systeme, im Hinblick auf deren Besonderheiten zu unterstützen.

Dieses Gesamtziel wurde in drei Teilziele aufgeteilt. Das erste Teilziel ist die Entwicklung von Werkzeugen und Modellen für Wearable-Computing-Arbeitssituationen als Basis für die Dokumentation und Kommunikation unter Beteiligten. Das zweite Teilziel ist die Entwicklung von Werkzeugen und Modellen, die den Designer bei der Auswahl und Konfiguation geeigneter Interaktionsgeräte unterstützen. Das dritte Teilziel ist schließlich die Berücksichtigung der gegenseitigen Beeinflussung verschiedener Interaktionsgeräte miteinander und mit der Arbeitskleidung des Benutzers (*Tragbarkeit*) sowie die gleichzeitige Durchführung verschiedener Tätigkeiten (*Multitasking*).

Wissenschaftliche Beiträge der Arbeit und Evaluation.
Der wissenschaftliche Rahmenbeitrag der Arbeit ist ein benutzerorientierter Prozess zur Unterstützung des Entwurfs von Wearable-Computing-Systemen. Dieser Prozess wird durch Werkzeuge und drei Modelle unterstützt: dem *Arbeitssituationsmodell*, dem *Benutzermodell* und dem *Computersystemmodell*. Ein Beispiel ist das im Computersystemmodell enthaltene *Interaktionsgerätemodell*, das über den Stand der Forschung hinaus geht, indem es nicht nur die technischen Merkmale eines Interaktionsgerätes beschreibt, sondern auch dessen Anforderungen an den Benutzer.

Ein weiterer Beitrag ist die *Simulation einer Arbeitssituation*, die auf den genannten Modellen aufsetzt und so die Kompatibilität eines Interaktionsgerätes mit einer gegebenen Arbeitssituation automatisch zu bestimmen gestattet.

Die Evaluierung der Arbeit ist zweigeteilt. Im ersten Schritt werden die entwickelten Modelle theoretisch mit denen der nächstverwandten Arbeit verglichen. Der zweite Teil ist eine praktische Evaluation der Arbeit. Hier werden zwei im Rahmen der Arbeit durchgeführte Fallstudien von Wearable-Computing-Projekten beschrieben und verglichen. Eines der Projekte verwendete einen herkömmlichen benutzerorientierten Entwurfsprozess und das andere den in der vorliegenden Arbeit entwickelten.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Die Arbeit mit Computern ist schon lange fester Bestandteil einer jeden Bürotätigkeit. Außerhalb von Büroumgebungen werden Computer bisher nur wenig eingesetzt. Mit zunehmendem technischen Fortschritt finden sie jedoch auch hier Einzug. Einen Anfang stellen hier PDAs und Smartphones dar. Diese Geräte ermöglichen ihrem Benutzer auch unterwegs auf Kommunikationsdienste und Anwendungen zuzugreifen, die vorher nur im Büro zugänglich waren. Beispiele für Kommunikationsdienste sind Email- und Kalenderanwendungen, während Systeme für Kundenbeziehungsmanagement typische mobile Anwendungen darstellen. Diese Art der mobilen Anwendungen ermöglicht es Büroangestellten, auch unterwegs mit ihren Daten zu arbeiten. Der Benutzer von Computern ist also immer noch derselbe. Möchte man neue Benutzerkreise erschließen, ist es notwendig in Bereiche vorzudringen, die derzeit nicht oder nur wenig mit Computern arbeiten. Überwiegend manuelle Tätigkeiten wie die Wartung von Flugzeugen sind hier Anwendungsgebiete.

Diese Art von Arbeitssituationen lässt sich wie folgt charakterisieren. Im Gegensatz zu Büroangestellten hat der mobile Benutzer eine Primäraufgabe in der "realen Welt", die nicht allein mit dem Computer durchgeführt werden kann. Diese Primäraufgabe ist häufig manuell. Die Verwendung eines Computers ist in solchen Fällen also sekundär und muss die Primäraufgabe unterstützen, um einen Nutzen zu erbringen. Dabei darf sie die Primäraufgabe jedoch nicht behindern, muss sich also in die Primäraufgabe integrieren. Diese Art der beiläufigen Unterstützung mit Hilfe von Computern, ist durch die fortschreitende Verbesserung und Miniaturisierung von Rechnern und Interaktionsgeräten in den letzten Jahren erst möglich geworden. Da eine solche Unterstützung in der Regel die Verwendung von am Körper getragenen Geräten einschließt, wird ein solches Computersystem auch als Wearable Computer bezeichnet.

Typische Anwendungsszenarien, die in der Forschung behandelt werden, sind Wartungsarbeiten, wie zum Beispiel die an Flugzeugen[BLCB06], Untersuchungs- und Operationsunterstützung im Gesundheitswesen [DHPS06, CKZ+07] sowie die Unterstützung von Feuerwehr und Polizei im Katastrophenschutz [BBR06, HB99]. Ein Szenario aus dem wearIT@work-Projekt [wea04b] soll an dieser Stelle die Umstände verdeutlichen, unter denen Wearable Computer eingesetzt werden.

Abbildung 1.1: (links) Typischer Endoskopieraum mit Patientenliege, Endoskopturm und Arbeitsplatz für Ärzte und Schwestern. (rechts) Bedienteil eines Endoskops mit Steuerrädern um die Kamera zu bewegen und Knöpfen für Zusatzfunktionen wie Fotos.

1.1 Beispiel: Endoskopie

Im Folgenden wird die derzeitige Arbeitssituation einer Endoskopieuntersuchung in der chirurgischen Ambulanzabteilung eines Krankenhauses erläutert, um zu zeigen wie solche Szenarien von Wearable-Computing-Lösungen profitieren können.

Bei einer Endoskopieuntersuchung wird eine Kamera mithilfe eines flexiblen Schlauchs in den Patienten eingeführt, um entweder den Magen (Gastroskopie) oder den Darm (Coloskopie) von innen zu untersuchen. Beide Untersuchungen werden sowohl zur Vorsorge als auch nach einem chirurgischen Eingriff durchgeführt. In einer Ambulanzabteilung herrscht der für Krankenhäuser typische Zeitdruck, da zu geplanten Untersuchungen häufig noch Notfälle hinzukommen. Untersuchungen werden von einem Arzt und ein bis zwei Pflegern durchgeführt.

Eine Untersuchung läuft dabei immer nach folgendem Schema ab. Zunächst werden Raum und Endoskop auf die Untersuchung vorbereitet. Ein steriles Endoskop wird geholt und benötigte Verbrauchsmaterialien und Medikamente werden aufgefüllt. Sobald der Patient eintrifft, wird dieser über die Untersuchung aufgeklärt. Der Arzt trifft meist erst jetzt ein, da er direkt von einer anderen Untersuchung im Nebenzimmer kommt. Er muss sich zunächst am Computer über den Patienten und die geplante Untersuchung informieren. Anschließend spricht er mit dem Patienten, um sich nach dessen Wohlbefinden zu erkundigen und gegebenenfalls erneut die Untersuchung zu erläutern.

Ist alles vorbereitet, beginnt die eigentliche Untersuchung. Arzt und Assistenten ziehen sterile Kittel und Handschuhe an, um eine Übertragung von Krankheitserregern zu vermeiden. Der Arzt steuert die Kamera mithilfe der Steuerrädchen am Bedienteil des Endoskops und beobachtet gleichzeitig das Kamerabild auf dem Monitor(siehe Abb.1.1). Hierfür benötigt er beide Hände. Gleichzeitig spricht er mit dem Patienten, um diesen direkt über den Status der Untersuchung zu informieren und sich nach dessen Wohlbefinden zu erkundigen. Außerdem gibt er Anweisungen an die Assistenten. Einzelne Kamerabilder können bei Bedarf zur Dokumentation ausgedruckt werden. Zusätzlich ermöglicht das Endoskop die Entnahme von Gewe-

Abbildung 1.2: (links) Dokumentation einer Untersuchung. Fotos, Probenbeschreibung und Kurzbericht. (rechts) Endoskopieturm mit (von oben nach unten) Monitor, Steuereinheit, Tastatur und Drucker.

beproben und die Verabreichung von Injektionen. Um die Bewegung der Kamera zu erleichtern, kann Luft durch das Endoskop gepumpt oder abgelassen werden. Hat der Arzt alle benötigten Informationen und Proben gesammelt, wird das Endoskop entfernt.

Assistent und Arzt ziehen nun Handschuhe und Kittel wieder aus und beginnen mit der Nachbereitung der Untersuchung. Die Assistenz kümmert sich um den Patienten sowie die verwendeten Geräte, etikettiert Proben und Fotos. Der Arzt muss nun den Verlauf der Untersuchung sowie deren Ergebnis dokumentieren. Hierzu muss er Position und Inhalt jedes Fotos und jeder Probe festhalten. Der Verlauf der Untersuchung und verwendete Maßnahmen werden auf einem Formblatt dokumentiert. Anschließend schreibt er zunächst einen Kurzbericht auf Papier, falls die Ergebnisse dringend benötigt werden. Danach erfolgt das Befunddiktat, welches später in der Schreibstube in den Computer getippt wird (siehe Abb.1.2).

Alles in allem dauert eine solche Untersuchung zwischen 40 und 75 Minuten, wobei etwa 10 bis 45 Minuten auf die eigentliche Arbeit mit dem Endoskop entfallen. Die lange Zeit, die für die eigentliche Untersuchung aufgewendet wird, birgt ein enormes Fehlerpotenzial. Da der Arzt aufgrund der Hygienebestimmungen während der Untersuchung nicht in der Lage ist den Computer zu bedienen, muss er sich alle relevanten Informationen aus der Patientenakte über den Verlauf der Untersuchung hinweg merken. Gleichzeitig muss er sich auch an die Informationen erinnern, die er für die Dokumentation der Untersuchung und für die Zusammenstellung des Befunds benötigt. Insgesamt bedeutet dies, dass die Gefahr ein wichtiges Detail zu vergessen oder Informationen zu vertauschen sehr groß ist, was fatale Folgen für den Patienten

haben kann. Dieses Problem könnte reduziert werden, wenn der Arzt auch während der Untersuchung in der Lage wäre, mit dem Computer zu interagieren. Fehlende Informationen, wie z. B. Vergleichsbilder aus vorherigen Untersuchungen, könnte er gleich einsehen und wichtige Erkenntnisse sofort dokumentieren.

Die manuelle Primäraufgabe ist in diesem Fall die Endoskopieuntersuchung, während Einsicht und Dokumentation von Informationen die sekundäre Computeraufgabe darstellen. Im Verlauf dieser Arbeit wird dieses Szenario immer wieder als Beispiel dienen, um die vorgestellten Methoden zu verdeutlichen.

1.2 Ziele

Der Entwurf von Wearable-Computing-Anwendungen ist immer noch schwierig. Die Verzahnung von Primäraufgabe und Wearable-Computing-System führt dazu, dass sich der Entwurf eines solchen Systems in einigen Punkten erheblich vom Entwurf herkömmlicher Benutzungsschnittstellen unterscheidet. Das *Gesamtziel* dieser Arbeit ist es daher den Entwurfsprozess für Wearable-Computing-Systeme, gerade im Hinblick auf deren Besonderheiten, zu unterstützen. Dieses Gesamtziel wurde nach einer Analyse des State of the Art in drei Teilziele unterteilt, die nun vorgestellt werden.

Teilziel 1: Entwicklung von Werkzeugen und Modellen zur Dokumentation und Kommunikation einer Wearable-Computing-Arbeitssituation.
Das erste Ziel ist es, die Dokumentation und Kommunikation der Arbeitssituation des Benutzers zu unterstützen. Im Gegensatz zu den meisten Büroanwendungen, ist die Arbeitssituation zu Beginn des Entwurfsprozesses nicht bekannt. Bei Büroanwendungen können Designer und Entwickler meist auf ihre eigenen Erfahrungen mit der Arbeitssituation zurückgreifen, da sie selbst in einem Büro arbeiten. Bei Wearable Computing Szenarien ist dies in der Regel nicht möglich. Deshalb ist es notwendig die Arbeitssituation des Benutzers und dessen Primäraufgabe so zu dokumentieren, dass sich alle Teammitglieder ein Bild machen können für welches Szenario sie entwerfen und entwickeln. Nur so kann sichergestellt werden, dass die Arbeitssituation beim Entwurf auch angemessen berücksichtigt werden kann.

Teilziel 2: Entwicklung von Werkzeugen und Modellen zur Unterstützung bei der Auswahl geeigneter Interaktionsgeräte
Das zweite Ziel ist es, die Auswahl geeigneter Interaktionsgeräte zu unterstützen. Für Büroanwendungen wird in der Regel eine Kombination aus Maus und Tastatur verwendet, die in Einzelfällen um ein Spezialgerät, z. B. einem Grafiktablett für Grafiker, erweitert wird. Einen solchen allgemeingültigen Standard gibt es für Wearable-Computing-Systeme nicht und kann es auch nicht geben, da jede Arbeitssituation eigene Anforderungen an die verwendeten Interaktionsgeräte stellt. Es ist also am Designer eine geeignete Gerätekombination zu finden, die flexibel genug ist, um dem Benutzer die Durchführung seiner Computeraufgabe zu ermöglichen, ohne seine Primäraugabe zu behindern. Eine Entwurfsunterstützung sollte dem Designer die Möglichkeit bieten, geeignete Interaktionsgeräte zu identifizieren und mit der Computeraufgabe zu verknüpfen.

Teilziel 3: Berücksichtigung der Aspekte Tragbarkeit von Interaktionsgeräten und Multitasking-Situationen
Die Werkzeuge und Modelle die für die ersten beiden Ziele entwickelt werden müssen sollten die Aspekte der Tragbarkeit von Interaktionsgeräten, sowie Multitasking-Situationen berücksichtigen. Diese beiden Wearable-Computing-spezifischen Aspekte werden derzeit von keiner der identifizierten Vorarbeiten beachtet.

- Ein Wearable-Computing-System soll parallel zur Primäraufgabe des Benutzers verwendet werden, also sogenanntes *Multitasking* erlauben. Dies ist jedoch nur möglich, wenn die Interaktionsgeräte nur solche Ressourcen des Benutzers verwenden, die er nicht bereits für seine Primäraufgabe benötigt. Die Entwurfsunterstützung sollte also die Möglichkeit bieten, Interaktionsgeräte nach diesem Kriterium zu bewerten und auszuwählen.

- Meist beinhalten Wearable-Computing-Systeme Interatkionsgeräte, die am Körper getragen werden und teilweise sogar in die Kleidung integriert sind. Deshalb muss beim Entwurf darauf geachtet werden, dass die Interaktionsgeräte im Rahmen der betrachteten Arbeitssituation auch *tragbar*, im Sinne von anziehbar sind. Man kann davon ausgehen, dass Faktoren wie Gewicht, Abwärme oder Feuchtigkeitstransport bereits beim Entwurf spezieller Wearable-Interaktionsgeräte berücksichtigt wurden. Deshalb spielt für die *Tragbarkeit* vor allem die Kompatibilität mit der Arbeitskleidung des Benutzers und seiner Primäraufgabe eine Rolle. Natürlich muss auch darauf geachtet werden, dass die verwendeten Interaktionsgeräte auch untereinander kompatibel sind.

Der Benutzer und seine Arbeitssituation stehen bei allen diesen Wearable-Computing-spezifischen Problemen im Mittelpunkt. Deshalb werden in Wearable-Computing-Projekten häufig benutzerorientierte Entwurfsprozesse (engl. user centered design) eingesetzt. Die Methoden und Werkzeuge, die diese Prozesse bereitstellen, sind jedoch nicht ausreichend, um die eben genannten Ziele zu erreichen. Vor diesem Hintergrund ist das Gesamtziel dieser Arbeit, den Entwurf von Wearable-Computing-Systemen in den genannten Punkten zu unterstützen. Die entwickelte Unterstützung sollte sich dabei in bestehende, benutzerorientierte Entwicklungsprozesse eingliedern.

1.3 Beiträge

Die vorliegende Dissertation leistet die folgenden wissenschaftlichen Beiträge:

- Erstens, einen *Entwurfsprozess*, der den Entwurf von Wearable-Computing-Systemen, besser und umfangreicher unterstützt, als die nächsten verwandten Arbeiten. Der Entwurfsprozess stellt Modelle und Werkzeuge zur Verfügung, um Wearable-Computing-spezifische Entwurfsaspekte besser zu unterstützen. Die Modelle erlauben die Modellierung einer Wearable-Computing-Arbeitssituation, wobei zwischen der Primäraufgabe und der Computeraufgabe des

Benutzers unterschieden wird. Diese Modellierung dokumentiert die Arbeitssituation und erleichtert so die Kommunikation mit Teammitgliedern, die keinen Kontakt mit dem Benutzer haben. Die konsequente Beachtung der zeitlichen Abfolge einer Arbeitssituation, ermöglicht erstmals die Interaktion des Wearable-Computing-Systems für bestimmte Situationen innerhalb einer Arbeitssituation zu optimieren.

- Zweitens, ein *Interaktionsgerätemodell*, das sowohl technische als auch ergonomische Aspekte eines Interaktionsgerätes beschreibt. Die technische Beschreibung erlaubt es herauszufinden, ob eine Menge von Interaktionsgeräten ausreicht um eine Computeraufgabe zu erledigen. Dazu werden *Interaktionsstrategien* als Bindeglied verwendet, die von konkreten Interaktionsgeräten abstrahieren. Die ergonomischen Aspekte hingegen beschreiben die Interaktionsressourcen, die der Benutzer benötigt um das Gerät zu verwenden. Dabei wird zwischen dem passiven Tragen des Gerätes und der aktiven Bedienung unterschieden.

- Drittens, eine *Simulation der Arbeitssituation*, die eine parallele Betrachtung der Auswirkungen von Arbeitssituation und Interaktionsgeräten auf den Benutzer erlaubt. Diese Simulation verwendet die ergonomischen Aspekte des Interaktionsgerätemodells um die Kompatibilität des Geräts mit der Arbeitssituation automatisch zu ermitteln. Die Kompatibilität wird dabei, abhängig vom Zeitpunkt innerhalb einer konkreten Arbeitssituation, ermittelt. Verwandte Vorarbeiten verwenden hierfür lediglich statische Modelle, die nur sehr einfache Aussagen über die Kompatibilität zulassen.

- Viertens, den Entwurf eines *Werkzeugs zur computergestützten Sequenzanalyse einer Arbeitssituation*, das auf die Bedürfnisse bei der Beobachtung von Wearable-Computing-Szenarien optimiert wurde. Durch eine Benutzerstudie wurde weiterhin der Zeitgewinn einer computergestützten Sequenzanalyse und der damit verbundene Qualitätsverlust, im Vergleich mit einer Videoanalyse, quantifiziert.

1.3.1 Publikationen

Einige Aspekte dieser Arbeit wurden als Forschungsbeiträge in Tagungsbänden von Informatik Konferenzen veröffentlicht. Die spezifischen Probleme, die bei der Entwicklung von Wearable-Computing-Systemen auftreten, werden in [KZZC07] diskutiert. Das Benutzermodell sowie Teile des Interaktiongsgerätemodells und der Algorithmus zur Analyse der Kompatibilität von Interaktionsgeräten wurde in [KM07a] veröffentlicht. Das Konzept des TaskObserver wurde in [Klu07] vorgestellt und in [KM07b] evaluiert. Außerdem beschreiben weitere Veröffentlichungen den benutzerorientierten Entwurfsprozess, der dieser Arbeit zugrunde liegt [CKZZ06, CKZ+07, ABK+08].

1.4 Aufbau und Struktur der Arbeit

Die vorliegende Arbeit ist folgendermaßen strukturiert. In **Kapitel 2** werden zunächst drei Definition des Begriffs Wearable Computing vorgestellt. Aus diesen wird anschließend eine Arbeitsdefinition entwickelt. Danach werden typische Wearable-Computing-Hardware sowie Wearable-Computing-Szenarien analysiert, die im Folgenden als Referenzgrundlage dienen. Ferner werden Wearable-Computing-spezifische Entwurfsprobleme erläutert. Aus diesen ersten Abschnitten werden Anforderungen extrahiert, auf Grund derer vier Vorarbeiten bewertet werden. Aus diesen werden dann die oben genannten Teilziele abgeleitet.

In **Kapitel 3** werden die Konzepte und Modelle vorgestellt, die zur Umsetzung dieser Arbeit entwickelt wurden. Auf Basis dieser Konzepte wird anschließend ein dazugehöriger Prozess beschrieben, der bestehende benutzerorientierte Entwicklungsprozesse um Lösungen von Wearable-Computing-spezifischen Problemen ergänzt.

In **Kapitel 4** werden die Modelle und Algorithmen der Unterstützung im Detail beschrieben. Des Weiteren wird kurz auf die Implementierung der Komponenten und Werkzeuge eingegangen.

In **Kapitel 5** werden die entwickelten Werkzeuge und Modelle hinsichtlich der in Kapitel 2 definierten Teilziele evaluiert. Dazu erfolgt zunächst ein theoretischer Vergleich der Modelle mit denen der nächstverwandten Arbeit. Danach werden die Teilziele anhand zweier Wearable-Computing-Projekte praktisch evaluiert. Eines dieser Projekte verwendete die hier beschriebenen Modelle und Werkzeuge, das andere nicht. Schließlich wird noch eine Benutzerstudie beschrieben, die den TaskObserver evaluiert. Diese Studie vergleicht außerdem wie groß Zeitgewinn und Qualitätsverlust einer computergestützten Sequenzanalyse im Vergleich zu einer Videoanalyse sind.

Kapitel 6 schließlich fasst die wissenschaftlichen Beiträge der Arbeit zusammen und gibt einen Ausblick auf mögliche Folgearbeiten.

Kapitel 2

State of the Art

In diesem Kapitel wird ein Überblick über den aktuellen Forschungsstand im Bereich des Wearable Computing[1] gegeben. Zunächst werden verschiedene Definitionen des Begriffs beleuchtet und die Wearable-Computing-Definition dieser Arbeit erläutert. Anschließend werden einige Wearable-Computing-Szenarien beschrieben und Beispiele typischer Ein- und Ausgabegeräte dargestellt. Zuletzt werden spezifische Aspekte der Entwicklung von Wearable-Computing-Systemen aufgezeigt. In den einzelnen Abschnitten werden zudem Anforderungen an eine Entwurfsunterstützung für Wearable-Systeme formuliert, aus denen die drei eingangs beschriebenen Teilziele dieser Arbeit abgeleitet werden.

2.1 Definition von Wearable Computing

Wearable Computing kann mit tragbarer (im Sinne von anziehbarer) Verarbeitung übersetzt werden. Obwohl dieser Begriff zunächst eine bestimmte Form der Hardware nahelegt, bezeichnet er eher ein Paradigma, als eine Art Computersysteme zu bauen. Obwohl es verschiedene Definitionen gibt, kann die erste Implementierung eines Wearable Computers in die 60er Jahre datiert werden [Rho98]. Ein von E.Thorp entwickeltes Computersystem zur Vorhersage einer Roulettekugel musste so klein und unauffällig wie möglich sein. Es sollte am Roulettetisch eingesetzt werden ohne Aufmerksamkein zu erregen [Tho69, Tho98]. Aufgrund dieser Anforderungen wurde ein kleines Computersystem entwickelt, das durch unauffällige Schalter mit der Hand oder dem Fuß bedient werden konnte und das seine Ergebnisse per Kopfhörer an den Benutzer meldete. Das System wurde seiner Zeit jedoch noch nicht als Wearable-Computing-System bezeichnet.

Obwohl man sich einig ist, dass Wearable Computing als Paradigma und nicht als Hardwareplattform mit bestimmten Eigenschaften zu verstehen ist, gibt es verschiedene Herangehensweisen, um die Eigenschaften eines solchen Systems zu beschreiben. Im Folgenden werden drei Sichtweisen von *Steve Mann*, *Thad Starner* und dem *wearIT@work Projekt* vorgestellt. Anschließend werden diese drei Sichtweisen diskutiert und die hier verwendete Definition von Wearable Computing vorgestellt.

[1] Eine korrekte und knappe Übersetzung der englischen Begriffe „wearable computing“ und „wearable computer“ ins Deutsche ist nicht möglich. Deshalb werden im weiteren Verlauf die englischen Begriffe verwendet. Auf die Einführung einer deutschen Übersetzung wird zugunsten der Verständlichkeit verzichtet.

2.1.1 Steve Mann

„Wearable Computing ermöglicht eine neue Form der Mensch-Computer Interaktion durch einen kleinen am Körper getragenen Computer (z. B. ein benutzerprogrammierbares Gerät), das jederzeit eingeschaltet und daher bereit und zugänglich ist. In diesem Sinne unterscheidet sich das neue Rechnerparadigma von dem anderer tragbarer Geräte wie Laptop Computer und Minicomputer (PDAs). Die „allzeit bereit“ Eigenschaft führt zu einer neuen Form der Synergie zwischen Mensch und Computer, charakterisiert durch die langfristige Anpassung mittels Beständigkeit der Benutzerschnittstelle.“

Wearable Computing nach Steve Mann [Man98b][2]

Die von Mann beschriebene neue Form der Synergie zwischen Mensch und Computer schlägt sich in drei wesentlichen Operationsprinzipien nieder. Das erste dieser drei Prinzipien ist die *Beständigkeit*. Der Wearable Computer ist ständig ohne aufwendiges Einschalten oder Ausklappen verfügbar und kann dem Benutzer potentiell helfen. Das zweite Operationsprinzip ist die *Erweiterung* der Wahrnehmung des Benutzers. Dabei ist die Annahme, dass der Computer nicht die Primäraufgabe des Benutzers darstellt. Er erweitert vielmehr den Intellekt und die Wahrnehmungsfähigkeit des Benutzers, um die Durchführung dessen eigentlicher Primäraufgabe zu unterstützen. Außerdem dient der Wearable Computer der *Vermittlung* zwischen dem Benutzer und der Informationstechnik. Dabei stellt der Wearable Computer auch sicher, dass die Privatsphäre des Benutzers gesichert wird.

Diese drei Operationsprinzipien werden durch eine enge Kopplung von Mensch und Computer erreicht, die in Abbildung 2.1 dargestellt ist. Betrachtet man die dargestellte Zusammenarbeit von Mensch, Wearable Computer und der Umgebung, dann werden sechs Signalpfade ersichtlich, die zu den wesentlichen Zielen beim Entwurf von Wearable Computer Systemen führen.

Der Wearable Computer darf *nicht Besitz ergreifend* sein, also die Wahrnehmung des Benutzers nicht behindern, damit dieser sich jederzeit anderen Dingen widmen kann. Der Wearable Computer sollte den Benutzer in seinen Handlungsmöglichkeiten auch *nicht einschränken*. Der Benutzer kann also jederzeit mit seiner Umgebung interagieren ohne durch den Wearable Computer behindert zu werden. Des weiteren sollte der Wearable Computer jederzeit *beobachtbar* sein. Das bedeutet, dass der Wearable Computer immer vom Benutzer wahrgenommen und seine Aufmerksamkeit auf sich lenken kann, sofern dies erforderlich ist. Der Wearable Computer sollte zudem auch *kontrollierbar* sein, sodass der Benutzer jederzeit in der Lage ist, das Verhalten des Wearable Computers zu bestimmen und in dessen Handlungen einzugreifen. Als weiteres Ziel gilt nach Mann, dass der Wearable Computer *aufmerksam* sein sollte und die Umgebung des Benutzers und den Benutzer selbst, mithilfe von Sensoren erfassen können müsste. Das letzte Ziel schließlich ist ein *kommunizierender* Wearable Computer, der mit dritten Parteien (Systemen oder Menschen) kommunizieren oder die Kommunikation vermitteln kann.

[2] übersetzt vom Autor.

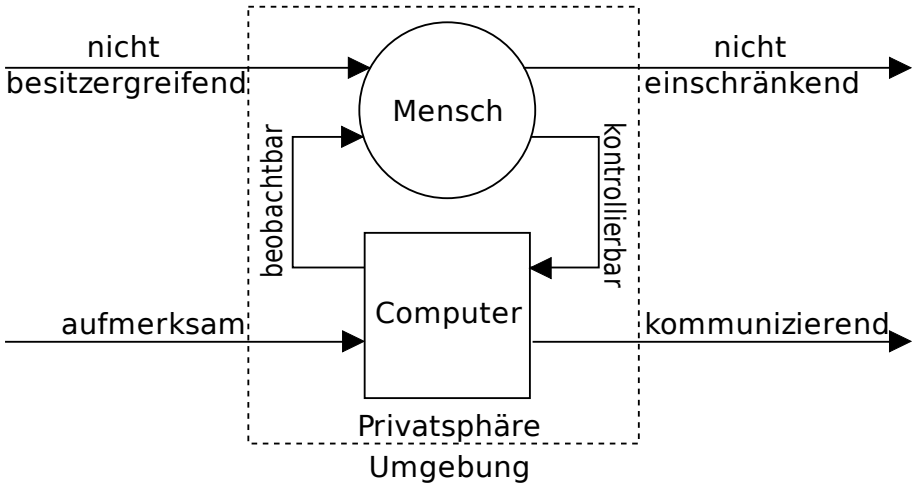

Abbildung 2.1: Die sechs Signalpfade die sich aus der engen Kopplung von Mensch und Wearable Computer ergeben [Man98a]. Zusätzlich ist die Hülle der Privatsphäre dargestellt, die durch den Wearable Computer beeinflusst werden kann. (Quelle [Man98a])

2.1.2 Thad Starner

„Wearable Computing verfolgt das Interface-Ideal eines unaufhörlich getragenen, intelligenten Assistenten, der Gedächtnis, Intellekt, Kreativität, Kommunikation sowie physische Sinne und Fähigkeiten erweitert.“

Wearable Computing nach Thad Starner [Sta01][3]

Aus dieser Definition des Wearable Computing leitet Starner vier Hauptfunktionen eines Wearable Computers ab. Die Erste ist die *ständige Verfügbarkeit und der Zugriff auf Informationsdienste*. Im Gegensatz zu anderen mobilen Geräten kann ein Wearable Computer also jederzeit und mit minimalen Kosten benutzt werden, um auf Informationsdienste zuzugreifen oder den Benutzer bei seiner Primäraufgabe zu unterstützen. Eine weitere Aufgabe ist das *Wahrnehmen und Modellieren von Kontext*. Der Wearable Computer sollte die Umgebung des Benutzers, den Benutzer selbst und seinen eigenen Zustand beobachten und modellieren, damit dieser jederzeit zur Verfügung steht. Durch die Modellierung des Kontextes kann der Wearable Computer dann die *Interaktionsmodalitäten an den Kontext anpassen*. Abhängig von der aktuellen Situation soll eine geeignete Art der Interaktion mit dem Benutzer ausgewählt werden. Die Bedienung des Wearable Computers ist dabei häufig sekundär in Relation zur Primäraufgabe des Benutzers. Als vierte Hauptfunktion soll der Wearable Computer die *Interaktion des Benutzers mit der Umgebung erweitern und vermitteln*. Beispielsweise sollte er zwischen dem Benutzer und Unterbrechungen wie Telefonanrufen vermitteln und stets Informationen, die für die aktuelle Situation relevant sind, sammeln und bereithalten.

Diese Hauptfunktionen unterscheiden sich wesentlich von der Funktionsweise herkömmlicher Computeranwendungen. Daher unterscheidet sich laut Starner nicht

[3] übersetzt vom Autor.

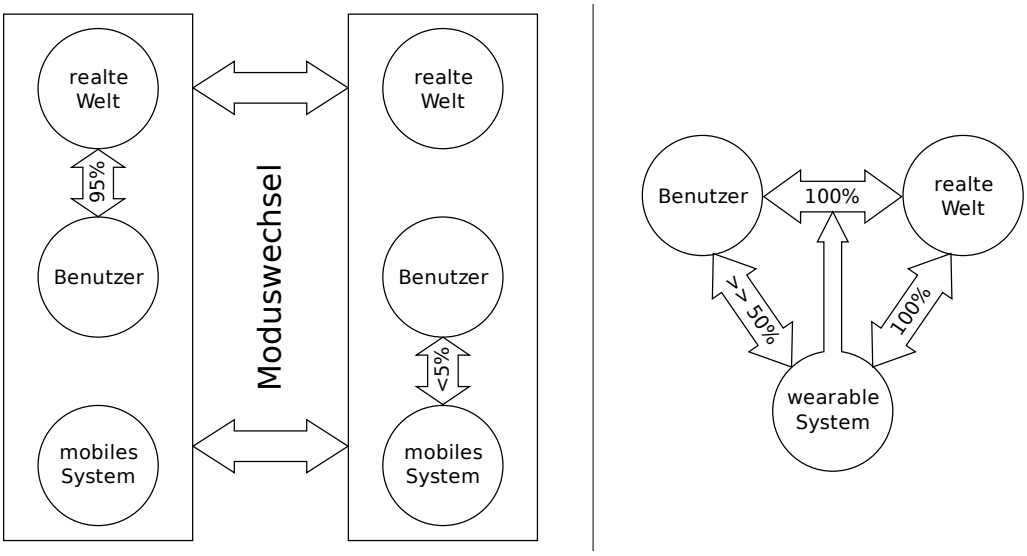

Abbildung 2.2: Unterschied zwischen (links)klassischen mobilen Systemen (PDA/Notebook/Mobiltelefon) und (rechts)Wearable-Computing-Systemen aus Sicht des wearIT@work Projekts. (Quelle [wea04a])

nur die Bedienung solcher Systeme, sondern auch deren Entwicklung erheblich. Da die Interaktion mit dem Computer nicht mehr die Primäraufgabe des Benutzers ist, sind bewährte Interaktionsparadigmen, wie z. B. das WIMP-Paradigma (engl. windows, icons, menus, pointer), hier nicht mehr anwendbar und neue Paradigmen müssen gefunden werden. Durch die Koexistenz von Primär- und Sekundäraufgaben des Benutzers entstehen zusätzlich *Multitasking* Situationen. Verschiedene Kombinationen von Interaktionsmodalitäten liefern in solchen Situationen unterschiedlich gute Ergebnisse. Deshalb rät Starner, die Wahl der Interaktionsmodalität sorgfältig im Einklang mit der Aufgabe des Benutzers und seiner Umgebung zu treffen [Sta02].

2.1.3 wearIT@work

Das wearIT@work Projekt[wea04b] definiert Wearable Computer über die Eigenschaft ständig verfügbar und vor allem nützlich zu sein [wea04a]. Heutige mobile Systeme sind häufig Adaptionen herkömmlicher Büroanwendungen und benötigen daher die vollständige Aufmerksamkeit des Benutzers. Deshalb muss der Benutzer sich zwischen der Interaktion mit dem mobilen System und der Interaktion mit seiner Umgebung entscheiden. Dies führt dazu, dass diese Systeme in der Regel nur etwa 1 % bis 5 % des Tages über nützlich sind. Wearable Computer hingegen sind permanent nützlich und können in vielen verschiedenen Umgebungen eingesetzt werden. Dabei wird im Rahmen des wearIT@work-Projektes immer von einer Nutzung im industriellen Kontext ausgegangen. Aufgrund des verwendeten Interaktionskonzeptes ist der Benutzer außerdem in der Lage, gleichzeitig mit System und Umgebung zu interagieren, wie Abbildung 2.2 verdeutlicht.

Aus Sicht des wearIT@work Projekts müssen für die Umsetzung dieses Konzeptes vier wesentliche Punkte berücksichtigt werden: Der erste Punkt ist die *Wahrehmung des Kontext* des Benutzers und damit auch dessen Umgebung mit Hilfe von Sensoren.

Dabei spielen zum einen Informationen über den Benutzer und dessen physischen und emotionalen Zustand eine Rolle, zum Anderen aber auch Informationen über die Umgebung, wie Temperatur, Luftfeuchtigkeit, Helligkeit und Position.

Der zweite Punkt ist die *Bedienung der Benutzerschnittstelle*, die *mit minimalem kognitivem Aufwand* ermöglicht werden muss, häufig auch mit eingeschränkten Modalitäten (z. B. ohne die Hände). Eine niedrige kognitive Last kann durch den Einsatz der zuvor gesammelten Kontextinformationen erreicht werden, indem man diese Informationen verwendet, um die Interaktion an die Situation anzupassen und damit zu vereinfachen. Der Umgang mit eingeschränkten Modalitäten kann ermöglicht werden, indem eine vielzahl verschiedener Interaktionsmethoden wie Gesten- und Spracherkennung implementiert werden und parallel zur Verfügung stehen.

Drittens muss das Wearable-System in der Lage sein, die *verfügbaren Kontextinformationen zu nutzen*, um in Eigeninitiative Aufgaben durchzuführen und Informationen zu beschaffen, die dem Benutzer die Arbeit erleichtern könnten. Als Beispiel wird die automatische Beschaffung von Wartungsanleitungen genannt, die durch die räumliche Nähe zu einem Bauteil angestoßen wird.

Viertens muss eine *nahtlose Integration in die Arbeitskleidung* des Benutzers erreicht werden, um dessen physische Aktivitäten nicht zu behindern. Dabei sollten auch ästhetische Aspekte berücksichtigt werden, um die soziale Akzeptanz des Systems zu gewährleisten.

Als Ideal dient hier ein modernes Hörgerät. Es ist unauffällig, die meiste Zeit des Tages über nützlich und erfordert keinerlei explizite Interaktion. Seine Hauptaufgabe ist die Erweiterung der Wahrnehmung des Benutzers. Moderne Geräte sind sogar in der Lage diese Wahrnehmungsverbesserung an unterschiedliche Geräuschsituationen anzupassen.

2.1.4 Diskussion

Mann's Definition eines Wearable Computers bildet die Grundlage der beiden Definitionen von Starner und dem wearIT@work Projekt. Allen Definitionen gemein ist daher der Gedanke des *ständigen* Begleiters, der den Benutzer *erweitert* oder *ermächtigt* indem er den *Kontext* des Benutzers erfasst und auswertet. Betrachtet man allerdings das angestrebte Anwendungsgebiet verschiedener Wearable Computing Projekte, so lassen sich zwei Schulen des Wearable Computing erkennen.

Die erste Schule, vertreten durch Starner und Mann, hat es zum Ziel, den Wearable Computer als *universelles Werkzeug* zu etablieren, das in der Lage ist, seinem Benutzer während des gesamten Tages nützlich zu sein. Der Wearable Computer soll also alltagstauglich werden. Obwohl der technische Fortschritt die Anwendung von Wearable Computern im täglichen Leben für den Benutzer erheblich vereinfacht hat, ist eine höhere Verbreitung wegen Problemen mit der sozialen Akzeptanz derzeit noch in weiter Ferne.

Aus diesem Grund existieren in der Praxis deutlich mehr Projekte, welche die zweite Schule des Wearable Computing verfolgen. Hier wird mit Wearable-Computing-Technologien ein *spezielles Werkzeug* zur Lösung eines bestimmtes Problems entwickelt [BS99]. Bei diesen Projekten, z. B. wearIT@work, handelt es sich meist um Industrieszenarien, bei denen der Benutzer unterwegs Informationen benötigt, während er gleichzeitig eine klar definierte Primäraufgabe zu erledigen hat, die vom

Wearable Computer lediglich unterstützt wird. Die Komponente der sozialen Akzeptanz verschwindet zwar nicht vollständig [KZZC07], oft überwiegt jedoch der wirtschaftliche Nutzen einer solchen Anwendung, sodass der Einsatz von Wearable Computern möglich wird. Bei diesen Projekten liegt der Fokus daher auf der Integration von Primäraufgabe und Computerinteraktion im Rahmen einer Arbeitssituation. Im Gegensatz zur ersten Schule ist die Primäraufgabe in diesem Fall allerdings bekannt. So kann der Wearable Computer optimal an die verfügbaren Modalitäten angepasst werden. Genau solche Szenarien werden auch im Rahmen dieser Arbeit behandelt, indem der systematische Entwurf eines Wearable-Systems, unter Berücksichtigung einer Primäraufgabe, unterstützt wird.

2.1.5 Arbeitsdefinition: Wearable Computing

Ausgehend von der Annahme, dass Wearable Computing in dieser Arbeit stets als spezielles Werkzeug zur Lösung eines bestimmten Problems verwendet wird, lassen sich einige zentrale Begriffe klarer definieren.

Definition 1 (Wearable Computing)

Wearable Computing *ist die technische Unterstützung einer parallel zu der* Primäraufgabe *durchgeführten, sekundären* Computeraufgabe*, vor dem Hintergrund einer* Arbeitssituation*.*

Beispiele für Wearable Computing nach dieser Definition sind das bereits in Abschnitt 1.1 beschriebene Endoskopieszenario sowie die beiden in Abschnitt 2.3 beschriebenen Szenarien.

Definition 2 (Arbeitssituation)

Die Arbeitssituation *ist eine zeitlich zusammenhängende Tätigkeit einer Person, die überwiegend aus einer manuellen Tätigkeit besteht und in einer definierten Umgebung stattfindet.*

Die Arbeitssituation der Endoskopieuntersuchung entspricht beispielsweise der Durchführung einer Untersuchung aus Sicht der Ärzte, inklusive Vorbereitung und Dokumentation. Die Untersuchung findet in einem dedizierten Endoskopieraum statt.

Definition 3 (Primäraufgabe)

Die Primäraufgabe *beinhaltet all die Aspekte der Arbeitssituation, die geeignet sind, den Entwurf und die gleichzeitige Bedienung eines Computersystems zu beeinflussen.*

Definition 4 (Computeraufgabe)

Die Computeraufgabe *ist der Teil der Arbeitssituation, der in Zukunft mit Hilfe eines Wearable Computers durchgeführt werden soll.*

Im Falle des Endoskopieszenarios besteht die Primäraufgabe aus der Bedienung der Endoskopiegeräte sowie Umgebungseinflüssen wie Helligkeit und Lautstärke. Die

Computeraufgabe besteht hier aus der Einsicht der Patientenakte und der Dokumentation der Untersuchungsergebnisse. Die Eigenschaften eines Wearable Computers, wie sie von Mann, Starner und dem wearIT@work-Projekt gefordert werden, bleiben dabei weiterhin bestehen.

2.1.6 Anforderungen

Aus der vorgestellten Wearable-Computing-Definition dieser Arbeit sowie den geforderten Eigenschaften lassen sich eine Reihe von Anforderungen an eine Entwurfsunterstützung ableiten. Diese sind zunächst die *Berücksichtigung der Arbeitssituation*, die *Auswahl geeigneter Interaktionsgeräte*, der *Entwurf angepasster Benutzungsschnittstellen* sowie die *Kontextsensitivität* von Wearable-Systemen:

Anforderung A 1 (Berücksichtigung der Arbeitssituation)

Eine Entwurfsunterstützung sollte es ermöglichen, die Arbeitssituation des Benutzers zu berücksichtigen, um die Interaktion mit dem Wearable Computer in diese integrieren zu können.

Diese Anforderung lässt sich aus der Annahme ableiten, dass ein spezielles Werkzeug für eine bestimmte Arbeitssituation entworfen werden soll. Die Arbeitssituation besteht sowohl aus der Primäraufgabe des Benutzers, als auch aus dessen Computeraufgabe. In Büroumgebungen kann sich der Benutzer in der Regel zu 100% auf die Computeraufgabe konzentrieren. Hier muss er seine Aufmerkasmkeit jedoch zwischen Primär- und Computeraufgabe aufteilen. Um eine benutzbares Wearable-Computing-System entwerfen zu können müssen deshalb beide zusammen betrachtet werden.

Anforderung A 2 (Auswahl geeigneter Interaktionsgeräte)

Die Auswahl der Interaktionsgeräte sollte an die Arbeitssituation angepasst sein und die besonderen Bedingungen der Primäraufgabe berücksichtigen. Gleichzeitig müssen die Interaktionsgeräte aber auch die Computeraufgabe erfüllen können. Der Auswahlprozess geeigneter Geräte sollte deshalb durch die Entwurfsunterstützung vereinfacht oder automatisiert werden.

Bei einer Büroanwendung kann auf die bewährte Interaktionsgerätekombination Maus, Tastatur und Bildschirm zurückgegriffen werden. Eine parallele Primäraufgabe sorgt hingegen in den meisten Fällen dafür, dass eine andere Kombination von Interaktionsgeräten gefunden werden muss. Diese muss sowohl an die Computeraufgabe, als auch an die Primäraufgabe angepasst werden. Zudem muss der Designer aus einer Vielzahl verfügbarer Interaktionsgeräte auswählen, um eine optimale Lösung zu finden. Da es schwierig ist die Menge an möglichen Kombinationen sowie die Vor- und Nachteile jedes einzelnen Interaktionsgerätes gleichzeitig, im Blick zu behalten, sollte die Entwurfsunterstützung diesen Prozess vereinfachen oder teilweise automatisieren.

Anforderung A 3 (Entwurf angepasster Benutzungsschnittstellen)

Die Entwurfsunterstützung sollte den Entwurf von Benutzungsschnittstellen ermöglichen, die an die verwendeten Interaktionsgeräte und die Arbeitssituation angepasst sind.

Da die geeigneten Interaktionsgeräte für jede Arbeitssituation erneut ermittelt werden müssen, ist es auch notwendig die Benutzungsschnittstelle an diese Geräte anzupassen. Dabei muss zum Beispiel berücksichtigt werden in welcher Form und Menge dem Benutzer Informationen präsentiert werden. Diese Entscheidung hängt stark davon ab, wie sehr sich der Benutzer auf seine Primäraufgabe konzentrieren muss und welche Computeraufgabe er gleichzeitig zu erledigen hat.

Anforderung A 4 (Kontextsensitivität)

Eine Entwurfsunterstützung sollte den Entwurf und die Entwicklung von Anwendungen erleichtern, die zur Laufzeit auf die Umgebung und den Kontext des Benutzers reagieren können.

Alle drei vorgestellten Wearable-Computing-Definitionen sind sich einig, dass ein Wearable Computer in der Lage sein sollte seine Umgebung wahrzunehmen. Die so gewonnenen Informationen sollte er verwenden, um den Benutzer bei der Durchführung seiner Arbeit zu unterstützen. Er kann beispielsweise Teilschritte automatisch abarbeiten oder Auswahlmöglichkeiten einschränken. Um solch ein Verhalten zu erreichen, müssen jedoch zwei Dinge implementiert werden. Zunächst muss der Wearable Computer seine Umwelt über Sensoren erfassen und diese Rohdaten zu höherwertigen Informationen verarbeiten. Dann muss die Wearable-Computing-Anwendung diese Informationen auswerten, um den Benutzer unterstützen zu können. Da sowohl die Erfassung der Sensordaten, als auch deren Nutzung in der Anwendung von der Arbeitssituation abhängen, sollte eine Entwurfsunterstützung die Entwicklung dieser beiden Teile vereinfachen.

Im weiteren Verlauf dieses Kapitels werden zusätzliche Anforderungen abgeleitet, die schließlich zum Vergleich verschiedener Vorarbeiten herangezogen werden. Auf der Basis dieses Vergleichs werden dann Lücken in den bestehenden Arbeiten identifiziert und die bereits in der Einleitung genannten Teilziele abgeleitet. Zunächst folgt jedoch eine Analyse existierender Wearable-Computing-Hardware, eine Beschreibung typischer Szenarien sowie eine Diskussion verschiedener Wearable-Computing-spezifischer Entwurfsaspekte.

2.2 Wearable-Computing-Hardware

Da Wearable-Computing-Systeme an die unterschiedlichsten Situationen angepasst werden müssen, gibt es eine Vielzahl verschiedener Hardwarekomponenten mit jeweils spezifischen Einsatzgebieten sowie Vor- und Nachteilen. Da Wearable-Computing-Anwendungen erst am Beginn ihrer Kommerzialisierung stehen, gibt es noch sehr wenige kommerzielle Geräte. Die Forschung beschäftigt sich jedoch schon lange mit dem Entwurf mobiler Computersysteme, weshalb eine Vielzahl unterschiedlicher Technologien und Interaktionsmethoden zur Verfügung stehen. Dieses Kapitel gibt sowohl einen Überblick über die wichtigsten, verfügbaren, zentralen Recheneinheiten

Abbildung 2.3: Q-Belt Integrated Computer (QBIC)[ALO+04], Zypad WL1000 [zyp07] und OQO [oqo07]. (von links nach rechts)

für Wearable-Systeme, als auch über eine repräsentative Auswahl typischer Interaktionsgeräte in diesem Feld, die das Ergebnis einer umfangreichen Literaturrecherche ist. Die Liste der hier genannten Interaktionsgeräte dient im Folgenden als Referenz und eine Entwurfsunterstützung sollte in der Lage sein die unterschiedlichen Eigenschaften dieser Interaktionsgeräte zu berücksichtigen.

Anforderung A 5 (Abdeckung verschiedener Interaktionsgeräte)

Eine Entwurfsunterstützung sollte in der Lage sein, die individuellen Eigenschaften verschiedener Interaktionsgeräte zu unterscheiden und diese beim Entwurf zu berücksichtigen. Dabei dient die nachfolgende Liste im Rahmen dieser Arbeit als Referenz.

2.2.1 Recheneinheiten

Jedes Wearable-Computing-System benötigt eine zentrale Recheneinheit. Diese Recheneinheit führt die Anwendungen aus, koordiniert die Verwendung von Interaktionsgeräten und sammelt und verarbeitet Kontextinformationen. Da der Benutzer diese Komponente jederzeit mitführen muss, ergeben sich einige Faktoren, die für ihre Benutzbarkeit entscheidend sind. Neben Größe und Gewicht spielen vor allem die Batterielaufzeit und die Wärmeentwicklung eine große Rolle [Dun04]. Massenprodukte wie PDAs und Mobiltelefone, die nicht speziell für den Einsatz als Wearable Computer konzipiert wurden, erfüllen daher in der Regel nicht alle Anforderungen. Während die Forschung in der Vergangenheit von Einzelanfertigungen als zentrale Recheneinheit bestimmt wurde, sind inzwischen einige wenige kommerzielle Systeme auf dem Markt, die eine deutlich höhere Stabilität und Verfügbarkeit versprechen als die Forschungsprototypen. Im Folgenden werden exemplarisch die drei, in Abb. 2.3 gezeigten Systeme kurz beschrieben.

Der **Q-Belt Integrated Computer (QBIC)** [ALO+04] ist ein Kleinstcomputer, der in einen Gürtel integriert ist und so sehr unauffällig getragen werden kann. Dies macht ihn mit seiner langen Batterielebensdauer und geringer Wärmeentwicklung zu einem vorbildlichen Wearable Computer. Der QBIC kann über Bluetooth mit Sensoren und Interaktionsgeräten kommunizieren und eine VGA-Schnittstelle

ermöglicht den Anschluß eines Head-Mounted Displays. Die Leistung des integrierten Prozessors ist jedoch sehr begrenzt, was rechenintensive Anwendungen ausschließt.

Das **Zypad WL1000** [zyp07] der Eurotech Group ist ein Komplettsystem, das am Unterarm befestigt wird. Hauptelement des Zypad ist ein Touchscreen, der sowohl für die Darstellung von Informationen, als auch für die Interaktion verwendet wird. Zusätzlich befinden sich an der Geräteoberseite Knöpfe, um Funktionen direkt ansprechen zu können. Die Kommunikation mit externen Geräten ermöglicht das Zypad mithilfe von Bluetooth und WLAN. Beschleunigungs- und Lichtsensoren werden verwendet um Strom zu sparen, wenn das System gerade nicht verwendet wird.

Neben diesen Systemen, die speziell für den Einsatz als Wearable Computer entwickelt wurden, gibt es auch einige mobile Kleinstcomputer, die sich als zentrale Recheneinheit eignen. Der **OQO** [oqo07] ist ein solches System, das die Fähigkeiten eines vollständigen PCs auf kleinstem Raum vereint. Dadurch wird die Entwicklung von Anwendungen deutlich erleichtert. Aus diesem Grund verwenden einige Forschungsprojekte den OQO als technologische Basis. Im Einsatz muss das Gerät in einer Tasche am Gürtel getragen werden, was aufgrund der geringen Ausmaße jedoch selten ein Problem darstellt. Zu den Schnittstellen gehören neben einem VGA-Ausgang auch WLAN und Bluetooth-Funk, sowie USB-Host Schnittstellen.

2.2.2 Interaktionsgeräte: Ein- und Ausgabe

Neben der zentralen Recheneinheit muss ein Wearable-System zusätzlich mit Geräten ausgestattet werden, die dem Benutzer die Interaktion mit dem System ermöglichen. Je nach Kontext sind dafür jedoch unterschiedliche Modalitäten und Interaktionsmethoden notwendig. Deshalb haben Industrie und Forschung eine Reihe von Interaktionsgeräten entwickelt, die jeweils eine oder mehrere Interaktionsmethoden implementieren. Im Folgenden wird eine Auswahl solcher Interaktionsgeräte vorgestellt. Die Auswahl deckt alle wichtigen Interaktionsmethoden ab, die im Wearable Computing Anwendung finden. Sie ist das Ergebnis einer umfassenden Literaturrecherche der Konferenzreihen CHI, MobileHCI, ISWC und UIST, der Jahre 1999 bis 2007.

Der **Lightglove** [HH01] ist ein etwa armbanduhrgroßes Gerät, das unter dem Handgelenk getragen wird. Das Gerät schickt 5 Lichtstrahlen in Richtung der Finger des Benutzers. Unterbricht ein Finger einen solchen Strahl, wird das reflektierte Licht erkannt und ein Tastendruck wird registriert. Zusammen mit einem ebenfalls integrierten 2D-Bewegungssensor lässt sich auf diese Weise Text eingeben und ein Mauszeiger steuern.

Der **Twiddler** [Sta99] ist eine Einhandtastatur, die Akkorde verwendet um mit 23 Knöpfen eine komplette englische Tastatur abzubilden. Ein Akkord ist dabei eine Kombination von bis zu vier Knöpfen, die gleichzeitig mit den vier Fingern einer Hand gedrückt werden müssen. Der Daumen ist damit frei um einen TrackStick als Mausersatz zu bedienen. Texteingabe mit dem Twiddler ist effizienter als andere mobile Texteingabeverfahren, erfordert jedoch das Erlernen der Akkorde [LSP+04].

FreeDigiter [MAS04] ist ein Gerät, das am Ohr des Benutzers befestigt wird, ohne im Ohr sitzen zu müssen. Mithilfe eines Infrarot-Abstandssensors werden Finger gezählt, die am Ohr vorbei bewegt werden. Die Anzahl der erkannten Finger

kann in eine Funktion oder Auswahl umgewandelt werden. Um unbeabsichtigte Interaktionen zu vermeiden, ist ein Beschleunigungssensor integriert, der Kopfnicken erkennen kann und so die Gestenerkennung ein- und ausschaltet. Die Zahlen von 1 bis 20 können mit diesem System zuverlässig eingegeben werden.

Das **Gesture Pendant** [SAAG00] besteht aus einer Infrarot Kamera, die wie ein Anhänger um den Hals getragen wird. Das System ist in der Lage verschiedene Gesten zu erkennen, die mit der Hand vor der Kamera ausgeführt werden. Um die Erkennung zu verbessern, ist zusätzlich noch ein Infrarotemitter in den Anhänger integriert. Neben einfachen Gesten, die zum ein- und ausschalten von Geräten geeignet sind können auch kontinuierliche Gesten erkannt werden, die zum Beispiel verwendet werden können, um die Lautstärke einer Musikanlage zu regeln.

Der **Finger-Ring** [Fuk05] ist eine Mikrofon und Lautsprecher Kombination, die wie ein Ring am Finger getragen wird. Dabei wird das Audiosignal direkt über Knochenvibration auf den Finger und so ins Ohr übertragen. Dazu muss die Fingerspitze allerdings in das Ohr gesteckt werden. Das Gerät ist daher unauffällig und muss nicht permanent am Ohr getragen werden, wie dies bei einem herkömmlichen Headset der Fall ist. Um Kommandos wie Abheben und Auflegen zu ermöglichen, ist zusätzlich ein 1D-Beschleunigungssensor integriert. Dieser erkennt rhythmisches Klopfen des Zeigefingers auf den Daumen und löst so verschiedene Funktionen aus.

Der **Acceleration Sensing Glove** [PFHP99] ist ein Handschuh, dessen Fingerspitzen mit 2D-Beschleunigungssensoren ausgestattet sind. Ein zusätzlicher Sensor befindet sich auf dem Handrücken. Mithilfe der Erdbeschleunigung lassen sich statische Gesten, bestehend aus einem oder mehreren ausgestreckten Fingern, erkennen. Der Winkel des Handrückens zur Erdbeschleunigung kann zudem als Mausersatz verwendet werden, während abrupte Bewegungen der Fingerspitzen als Maustasten dienen.

Eine **Unterarmtastatur** [TTG98, fro07] wird am Unterarm des Benutzers befestigt. Da die Bedienung nun nur noch mit einer Hand erfolgen kann, ist die Anordnung der Tasten so verändert, dass mithilfe einer Umschalttaste zwischen zwei virtuellen Tastaturhälften umgeschaltet werden kann.

Ein **Touchpad** [TGM+99] kann an verschiedenen Orten (Unterarm, Oberarm, Oberkörper, Oberschenkel oder Hüfte) befestigt werden, um jederzeit zur Verfügung zu stehen. Die Bedienung erfolgt mit dem Finger und die Funktionen sind die einer Maus.

Textile Knöpfe [TMTP02] können direkt in Kleidungsstücke integriert werden. Die Knöpfe werden aus mehreren Stoffschichten aufgebaut. Wird Druck auf den Knopf ausgeübt, dann ändert sich der Abstand zwischen den Schichten und die Interaktion wird ausgelöst.

Magnetschalter [NOKK04] können an verschiedenen Körperpositionen zur berührungslosen Interaktion verwendet werden. Dabei sind zwei Teile nötig. Ein Reedkontakt, der beispielsweise an der Brust befestigt werden kann und ein Festmagnet, der am Handgelenk getragen wird. Bringt man den Festmagnet in die Nähe des Kontakts, dann schaltet dieser und ermöglicht so einfache Interaktionen.

Taktile Ausgabesysteme [TDTA03, tac07] sind eine weitere Form der unauffälligen Interaktion mit dem Benutzer. Taktile Aktuatoren können verschiedene Frequenzen, Wellenformen und Rhythmen verwenden, um eine Reihe von Nachrichten zu übermitteln. Dabei kann der Aktuator an verschiedenen Stellen in die

Kleidung integriert werden, z. B. in den Gürtel oder in Schulterpolster.

Das Gerät namens **GestureWrist** [Rek01] misst mithilfe von kapazitiven Sensoren, die in einem Armband untergebracht sind, die Form des Handgelenks. Diese verändert sich mit der Bewegung der Finger, da sich die zugehörigen Muskeln im Unterarm befinden. Durch diese Formveränderung lassen sich einfache Gesten erkennen. Ein Rotationssensor erlaubt außerdem die kontinuierliche Veränderung von Daten. Ein piezoelektrisches Element ermöglicht einfache taktile Rückmeldungen an den Benutzer.

Ein **Head-Mounted-Display (HMD)** [lum07, lit07] ist ein Miniaturbildschirm, der vor einem Auge des Benutzers positioniert wird und ihm so die Möglichkeit gibt ständig auf die dargestellten Informationen zugreifen zu können. Es gibt durchsichtige HMDs und undurchsichtige. Durchsichtige HMDs überlagern die reale Welt während der Benutzer bei undurchsichtigen HMDs am Bildschirm vorbei schauen muss. HMDs werden in der Regel an einer vorhandenen Brille befestigt oder an einem Stirnband.

Der **Winspect Handschuh** [BNSS01] integriert drei Magnetschalter an Fingerspitzen und Daumen mit einem Neigungssensor, einem lokalen Ultraschall Positionierungssystem und einem RFID-Lesegerät. Der Neigungssensor und die Magnetschalter können verwendet werden, um eine Anwendung zu steuern. Hingegen werden die Ultraschall Positionierung und der RFID-Leser verwendet, um die Interaktion durch zusätzliche Kontextinformationen zu vereinfachen.

2.3 Wearable-Computing-Szenarios

Wearable Computing Systeme können in einem breiten Spektrum von Anwendungsszenarien nützlich sein. Im folgenden Abschnitt werden, zusätzlich zu dem in der Einleitung beschriebenen Endoskopieszenario, ein Visitenszenario und ein Szenario aus der Automobilproduktion exemplarisch herausgegriffen. Dies verdeutlicht wie Wearable-Computing-Lösungen konkret in Projekten zur Verbesserung einer Arbeitssituation eingesetzt wurden.

2.3.1 Gesundheitswesen

Im Gesundheitswesen gibt es eine Vielzahl von Einsatzszenarien für Wearable Computer. Neben dem vorgestellten Szenario der Endoskopieunterstützung auch die Unterstützung der mobilen Krankenpflege [DHPS06] oder dem Pflegepersonal in Krankenhäusern [NOKK04]. Ein weiteres Szenario, das im Rahmen des wearIT@work Projektes intensiv untersucht wurde, ist die Visite in einem Krankenhaus [CKZZ06, CKZ+07, KZZC07]. Der verwendete Entwicklungsprozess und die entwickelte Lösung werden im Folgenden beschrieben.

Die Visite ist ein zentraler Prozess des klinischen Alltags. Hier besucht der zuständige Arzt zusammen mit den Krankenschwestern alle Patienten seiner Station. Er macht sich ein Bild über den aktuellen Gesundheitszustand, vergleicht diesen mit der Historie, überprüft die bestehende Behandlung und ordnet gegebenenfalls weitere Untersuchungen, neue Medikation oder die Entlassung an. Die Qualität der Visite ist von entscheidender Bedeutung, da Fehler gravierende Auswirkungen auf

die Gesundheit des Patienten haben können. Abhängig von der Art der Station steht der Arzt zudem bei der Visite unter einem enormen Zeitdruck (z. B. in der Chirurgie).

Die Ausgangssituation im untersuchten Krankenhaus war ein teilweise elektronisch unterstützter Visitenprozess. Die meisten Teile der Dokumentation standen bereits im Computer zur Verfügung. Einige Informationen wurden jedoch noch auf Papier dokumentiert. Möchte der Arzt während der Visite auf die Patienteninformationen zugreifen oder Entscheidungen dokumentieren, benötigt er Zugriff auf das Computersystem des Krankenhauses. Derzeit haben die Ärzte die Möglichkeit bei der Visite einen Laptop auf einem Wagen mitzuführen, um so beim Patienten auf relevante Daten zugreifen zu können. Die Bedienung des Computers ist jedoch zum einen nicht an eine mobile Arbeitssituation angepasst und zum anderen erfordert sie die Bedienung mit den Händen. Dies ist für den Arzt jedoch problematisch, da er häufig zwischen dem Patienten und dem Computer wechseln muss und jeder Wechsel ein Desinfizieren der Hände erfordert. Aus diesem Grund werden Befunde meist vor der Visite vom Pflegepersonal ausgedruckt und Entscheidungen zunächst durch die Pfleger auf Papier dokumentiert. Erst nach der Visite werden diese Informationen in das Computersystem übertragen.

Die *Primäraufgabe* des Arztes ist es in dieser Arbeitssituation also mit dem Patienten über dessen Befinden und den Behandlungsverlauf zu reden und ihn gegebenenfalls zu untersuchen. Gleichzeitig muss er sich mit dem Pflegepersonal abstimmen um den weiteren Ablauf zu planen. Die *Computeraufgabe* des Arztes ist es die Patientenakte einzusehen, um eine Entscheidungsgrundlage für die weitere Behandlung zu haben, und diese ordnungsgemäß zu dokumentieren.

Die Anforderungen an eine Wearable Computing Lösung waren also ein berührungsloses Bedienungskonzept und die Optimierung der benötigten Zeit für den Arzt bei gleichzeitiger hoher Qualität der eingetragenen Informationen. In einem benutzerorientierten Prozess nach ISO 13407 [ISO99] mit mehreren Prototypen und Benutzerstudien wurde ein System entwickelt, das diesen Anforderungen gerecht wird. Dieses sieht folgendermaßen aus:

Am Bett des Patienten ist ein Bildschirm befestigt, der während der Visite zur Darstellung der Patienteninformationen verwendet werden kann und der zu jeder anderen Zeit als Unterhaltungssystem für den Patienten fungiert. Der Arzt trägt einen Wearable Computer am Gürtel, der drahtlos mit einem RFID-Lesegerät am linken Handgelenk und einem Interaktionsarmband am rechten Handgelenk kommuniziert. Außerdem trägt er ein Headset, das ebenfalls per Bluetooth angebunden ist. Die dritte Komponente ist ein PDA, der vom Pflegepersonal mitgeführt wird. Zusätzlich tragen sowohl Pfleger als auch Arzt einen Sensor bei sich, der es ermöglicht den Abstand der beiden Personen zu bestimmen.

Die „neue“ Visite läuft mit dem entwickelten System wie folgt ab: *Der Arzt und die Pflegerin betreten das Patientenzimmer. Zunächst identifiziert der Arzt den Patienten berührungslos mithilfe des RFID-Lesers am linken Arm. Daraufhin schaltet der Bildschirm am Bett vom Unterhaltungsprogramm auf eine Ansicht der Patientenakte um. In dieser kann der Arzt nun mit Gesten der rechten Hand navigieren, die über den Beschleunigungssensor erfasst werden. Zwischendurch kann der Arzt beliebig oft mit dem Patienten in Kontakt kommen, da er für die Interaktion keine Gegenstände berühren muss. Möchte der Arzt eine Anweisung dokumentieren, führt*

er die rechte Hand zum Ohr und die Sprachaufzeichnung wird gestartet. Durch Absenken der Hand wird die Aufzeichnung gestoppt und die Audiodatei an den zentralen Server geschickt. Gleichzeitig wird die Pflegerin auf dem PDA gebeten Details zur Anfrage einzugeben, die ebenfalls im System gespeichert werden. Diese beiden Arbeitsschritte erfolgen jedoch asynchron, sodass der Arzt die Untersuchung ungestört fortsetzen kann.

2.3.2 Automobilproduktion

Ein weiteres Einsatzgebiet für Werable-Computing-Technologie ist die Produktion. Beispielhaft wird hier ein Szenario aus dem Projekt wearIT@work vorgestellt, das sich mit der Qualifizierung von Arbeitskräften beschäftigt [IMSS06, SLR+06, MKS+07]. Die Montage eines modernen Autos ist komplex und für jeden Arbeitsschritt werden Spezialwissen und Spezialwerkzeuge benötigt. Geschehen hier Fehler, die unbemerkt bleiben führt dies schnell zu hohen wirtschaftlichen Schäden. Aus diesem Grund müssen alle Arbeiter mit hohem Aufwand bestens geschult werden. In dem hier betrachteten Unternehmen geschieht die Schulung in zwei Stufen. Zunächst durchläuft der Arbeiter den theoretischen Teil der Ausbildung, der mithilfe von E-Learning vermittelt wird. Anschließend wird in einer sogenannten Lerninsel die praktische Durchführung der gelernten Arbeitsschritte geübt. Derzeit muss während des praktischen Teils jederzeit ein Meister anwesend sein, da nur er die korrekte Durchführung der Arbeitsschritte überwachen kann. An dieser Stelle kann Wearable Computing helfen, indem die Kontrolle und Anleitung des Arbeiters automatisiert wird.

Die *Primäraufgabe* des Benutzers ist hier die Durchführung verschiedener Montageschritte. Seine *Computeraufgabe* ist zum Einen die Kontrolle der Korrekten Durchführung seiner Arbeitsschritte. Zum Anderen kann er zusätzliches Lernmaterial einsehen, wenn er den aktuellen Arbeitsschritt nicht durchführen kann oder er einen Fehler gemacht hat.

Um diese Computeraufgabe zu implementieren wurde der Arbeiter mit einem Wearable-Computing-System bestehend aus OQO, HMD, Headset und Beschleunigungssensoren am Handgelenk ausgestattet. Zusätzlich wurde die Lernumgebung mit weiteren Sensoren versehen, um die korrekte Durchführung einer Montage zu überwachen.

Mit Hilfe von Verfahren des maschinellen Lernens ist der Wearable Computer in der Lage den aktuellen Prozessschritt zu erkennen und auf Fehler zu überprüfen. Tritt solch ein Fehler auf, kann der Benutzer informiert werden. Je nach Bedarf kann über das HMD zusätzlich die Dokumentation des Arbeitsschrittes eingesehen werden. So können Fehler eigenständig korrigiert werden. Der Eingriff durch den Meister ist nur dann notwendig, wenn unvorhergesehene Probleme oder Verständnisschwierigkeiten auftreten. Da alle Informationen über die Anzahl der Fehler, Anzahl der Wiederholungen und der aktuelle Prozessschritt elektronisch erfasst werden, kann der Meister von einem zentralen Computer aus die Schulung mehrerer Arbeiter gleichzeitig überwachen. Auf diese Weise können die entstehenden Kosten deutlich reduziert werden und die Arbeiter sind schneller auf ihren Arbeitsplatz vorbereitet.

2.4 Wearable-Computing-Entwurfsaspekte

Dieser Abschnitt beschäftigt sich mit verschiedenen Aspekten des Entwurfs von Wearable-Computing-Anwendungen. Dabei wird insbesondere auf die Unterschiede beim Entwurf von Wearable-Systemen im Vergleich zu herkömmlichen Anwendungen eingegangen.

Beim Entwurf von Wearable-Computing-Systemen gibt es dieselben Schwierigkeiten, die auch beim Entwurf herkömmlicher Benutzungsschnittstellen auftreten. Allerdings gibt es auch Aspekte, die erst durch die enge Verwebung der realen Welt mit dem Computer entstehen [KZZC07]. Einige dieser Aspekte betreffen den Entwurfsprozess als solchen (Benutzerstudien und Kommunikation der Arbeitssituation, andere die Benutzbarkeit der fertigen Lösung (Multitaskung, Tragbarkeit und Non-User). Diese fünf Aspekte werden nun kurz beschrieben.

2.4.1 Benutzerstudien: Im Labor oder Vor Ort?

Benutzerstudien sind notwendig um die Benutzbarkeit einer Anwendung realistisch evaluieren zu können. Im Gegensatz zu Büroanwendungen muss bei Wearable-Systemen jedoch entschieden werden, ob die Studie unter Laborbedingungen oder in einem realen Kontext stattfinden soll. Beide Vorgehensweisen haben Vor- und Nachteile, weshalb die Entscheidung abhängig vom aktuellen Stand des Projektes ist.

Untersuchungen haben belegt, dass Benutzerstudien, die im Labor durchgeführt werden schlechtere Ergebnisse liefern als jene im realen Kontext [KS04, DThC06]. Trotzdem werden meist Laborexperimente durchgeführt, da die Bedingungen im Feld nur schwer zu kontrollieren sind und so die tatsächlichen Ursachen für Probleme oft unerkannt bleiben. Außerdem ist eine Feldstudie nur mit erheblichem Aufwand und einem funktionierenden Prototypen möglich. Bei Laborexperimenten hingegen lassen sich die Bedingungen präzise kontrollieren und der Aufwand bleibt gering, da auch Attrappen (engl. mock-ups) verwendet werden können, falls kein funktionierender Prototyp verfügbar ist oder dieser noch nicht die gewünschte Qualität erreicht hat. In der Praxis wird abhängig vom Status des Projekts eine Kombination aus beiden Verfahren gewählt [BS05].

Bei der Evaluierung von Wearable-Systemen muss zusätzlich die Primäraufgabe berücksichtigt werden. Für ein Laborexperiment muss die Primäraufgabe in der Regel simuliert werden, da die Geräte nicht vom Arbeitsplatz entfernt werden können oder die Arbeit nicht unter Laborbedingungen durchgeführt werden kann (z. B. bei der Endoskopie). Arbeiten, die sich mit der Simulation einer Primäraufgabe zum Zwecke einer Benutzerstudie beschäftigen berichten, dass diese Vorgehensweise durchaus gute Ergebnisse liefert [WD06].

2.4.2 Tragbarkeit von Interaktionsgeräten

Der Formfaktor eines Wearable-Computing-Systems spielt eine deutlich größere Rolle für die Benutzbarkeit als bei anderen Anwendungstypen. Da das Gerät über lange Zeiträume direkt am Körper getragen wird, spielen beispielsweise Gewicht und Wärmeentwicklung eine große Rolle. Ob ein Gerät gut tragbar (engl. wearable) ist, hängt von sehr verschiedenen Faktoren ab. Relevant sind laut Dunne [Dun04] vor

allem Druck und Einengung, Textur, Wärmebilanz, Feuchtigkeitstransport und Bewegungsfreiheit.

Gemperle hat zu diesem Thema eine Studie der Tragbarkeit (engl. wearability) durchgeführt. Er hat untersucht an welchen Stellen des Körpers Geräte über längere Zeiträume befestigt werden können, ohne den Benutzer zu stören [GKS+98]. Gemperle entwickelte zunächst eine Liste von Kriterien, welche die Tragbarkeit sicherstellen. Mithilfe dieser Kriterien und umfangreichen Studien wurden dann 8 geeignete Positionen identifiziert.

Neben den physischen Aspekten haben auch kognitive Aspekte eine Bedeutung für die subjektive Tragbarkeit eines Gerätes. Der menschliche Körper gewöhnt sich z. B. an Kleidungsstücke und nimmt diese nach einiger Zeit als Teil seiner selbst und nicht mehr als Fremdkörper wahr. Wäre dies nicht der Fall, würden wir bei jeder Bewegung von unserer Kleidung abgelenkt. Dunne u. A. haben untersucht welche Kriterien erfüllt sein müssen, um diese Anpassung zu erreichen und somit unerwünschte Effekte auf die kognitiven Fähigkeiten des Benutzers zu vermeiden [DS07].

All diese Aspekte der Tragbarkeit sind für die Gebrauchstauglichkeit einer Wearable-Computing-Lösung von entscheidender Bedeutung und müssen daher beim Entwurf berücksichtigt werden.

Anforderung A 6 (Berücksichtigung von Wearability)

Eine Entwurfsunterstützung für Wearable-Systeme sollte in der Lage sein die Wearability eines Interaktionsgerätes zu berücksichtigen und in den Entwurfsprozess einfließen zu lassen.

2.4.3 Multitasking

Allport hat mit seinen Studien gezeigt, dass Menschen in der Lage sind, mehrere Aufgaben gleichzeitig durchzuführen [AAR72]. Allerdings hängt die Performanz stark von der Art der Aufgaben und von den verwendeten Modalitäten ab, wie auch Studien von Hazeltine belegen [HR06]. Sollen die Primäraufgabe und Computeraufgabe parallel stattfinden, ist es wichtig, diesen Umstand zu berücksichtigen. Um die Auswahl effizienter Modalitätenkombinationen zu vereinfachen, beschreibt Wickens ein vierdimensionales Ressourcenmodell [Wic02] bestehend aus: *Verarbeitungsstufe* (wahrnehmend/reaktiv), *Wahrnehmungsmodalität* (visuell/auditiv), *visuellem Kanal* (fokal/peripher) und *Verarbeitungskode* (räumlich/sprachlich). Seine These ist, dass Aufgaben die mehr gleiche Ressourcen benötigen schlechter zusammenpassen, als solche, die disjunkte Ressourcen verwenden.

Eine Analyse wie gut sich zwei reale Aufgaben miteinander kombinieren lassen ist mithilfe von Performance Modellen möglich. CPM-GOMS von John und Kieras [JK96] ermöglicht beispielsweise die theoretische Ermittlung der Durchführungszeit einer Aufgabe, die mit einem Mobiltelefon durchgeführt wird. Dabei wird die Primäraufgabe des Benutzers mit berücksichtigt [JK04]. Eine Methode zur Entwurfsunterstützung von Wearable-Systemen sollte die Auswirkung der Durchführung mehrerer Paralleler Aufgaben also berücksichtigen und deren Ermittlung unterstützen.

Anforderung A 7 (Berücksichtigung der Auswirkungen von Multitasking)

Eine Entwurfsunterstützung für Wearable-Systeme sollte in der Lage sein den Designer dabei zu unterstützen, die Auswirkungen mehrerer parallel durchgeführter Aufgaben aufeinander in seinem Entwurf zu berücksichtigen.

2.4.4 Non-User

In vielen Anwendungsgebieten greifen Wearable Computer deutlich stärker in das soziale Umfeld des Benutzers ein, als dies herkömmliche Anwendungen tun. Um diese Begebenheit besser in den Entwurfsprozess integrieren zu können, führen Herstad u. A. [HSvT00] den Begriff des Non-User ein, „einer Person oder Gruppe, welche die fragliche Technologie nicht direkt benutzt, aber auf irgendeine Art und Weise durch deren Benutzung beeinflusst wird“. Sie argumentieren, dass eben dieser Non-User in den Entwurfsprozess integriert werden muss um ein sozial akzeptiertes System entwerfen zu können. Im Gesundheitswesen sind Ärzte und Pfleger meist die Endanwender eines Wearable-Computing-Systems. Patienten hingegen sind die Non-User, da sie anwesend sind, wenn der Arzt das System einsetzt. Weil das Verhältnis zwischen Arzt und Patient hauptsächlich auf Vertrauen und dem persönlichen Kontakt beruht ist es unerlässlich zu prüfen, ob die Verwendung eines Wearable Computers dieses Verhältnis negativ beeinflusst. In dem bereits beschriebenen Visitenszenario haben Ärzte und Pfleger beispielsweise erhebliche Bedenken gegen die Verwendung eines HMD geäußert, da dieses den direkten Blickkontakt zwischen Patient und Arzt stören könnte. Wenn möglich sollte so eine Aussage allerdings durch den eigentlichen Non-User verifiziert werden, was häufig nicht möglich ist. Ein weiteres Beispiel ist die Verwendung von Spracheingabe für die Interaktion mit dem Computer. Obwohl sie für den Arzt einfach ist, kann für den Non-User der Eindruck entstehen der Arzt würde sich mehr mit dem Computer beschäftigen als mit ihm. Die Beispiele zeigen, dass der Non-User einen nicht unerheblichen Einfluss auf den Erfolg eines Wearable-Systems haben kann. Deshalb sollte einen entsprechende Entwurfsunterstützung die Berücksichtigung seiner Bedürfnisse unterstützen.

Anforderung A 8 (Berücksichtigung des Non-User)

Die Entwurfsunterstützung sollte Hilfsmittel bereitstellen, die den Designer dabei unterstützen die Bedürfnisse des Non-User zu erfassen und zu berücksichtigen.

2.4.5 Kommunikation der Arbeitssituation

Ein weiterer Aspekt, der beim Entwurf von Wearable-Systemen berücksichtig werden muss, ist die Kommunikation der Arbeitssituation zwischen Teammitgliedern. In Projekten mit mehreren Teammitgliedern werden nur einige wenige die Chance haben, den Benutzer vor Ort zu besuchen und sich ein eigenes Bild der Arbeitssituation zu machen. Die anderen Teammitglieder sind hingegen darauf angewiesen, diese Informationen von ihren Kollegen zu erhalten. Beim Entwurf klassischer Büroanwendungen ist dies nur ein geringes Problem, da die Arbeitsumgebung den Teammitgliedern weitestgehend aus ihrer eigenen Erfahrung bekannt ist. Die Arbeitssituation

eines Wearable-Computing-Szenarios hingegen unterscheidet sich deutlich von einer Büroumgebung, sodass nicht auf die eigenen Erfahrungen zurückgegriffen werden kann.

Die Arbeitssituation muss diesen Teammitgliedern also angemessen kommuniziert werden. Kommt es hier zu Missverständnissen, können die Benutzeranforderungen nicht korrekt berücksichtigt werden, was zu zusätzlichen und vermeidbaren Entwurfsiterationen führt. Erfahrungen haben gezeigt, dass vor allem zeitliche Aspekte einer Arbeitssituation entscheidend für den erfolgreichen Entwurf eines Wearable-Systems sind [KZZC07]. Aus diesem Grund sollte eine Entwurfsunterstützung dabei helfen die Einzelheiten der zu unterstützenden Arbeitssituation an alle Mitglieder des Entwurfsteams zu kommunizieren.

Anforderung A 9 (Kommunikation der Arbeitssituation)

Die Entwurfsunterstützung sollte in der Lage sein, die erhobenen Informationen über eine Arbeitssituation so zu visualisieren, dass diese Dritten leicht verständlich gemacht werden kann. Nur so kann sichergestellt werden, dass alle Teammitglieder dasselbe Verständnis über die Arbeitssituation haben.

2.5 Verwandte Arbeiten

Insgesamt wurden in den vorangegangenen Abschnitten 9 Anforderungen an eine Entwurfsunterstützung für Wearable-Systeme identifiziert, die in Tabelle 2.1 zusammengefasst sind. Auf Basis dieser Anforderungen können nun verwandte Arbeiten bewertet und eingeordnet werden. Die betrachteten verwandten Arbeiten ordnen sich in zwei Kategorien, der *Implementierungsunterstützung* sowie der *Entwurfsunterstützung*. Auf Grund des recht jungen Alters dieses Forschungsgebietes existieren nur wenige Arbeiten, die sich originär mit dem Thema Entwurf von Wearable-Computing-Lösungen beschäftigen. Aus diesem Grund wurden auch Arbeiten herangezogen, die trotz ihrer unterschiedlichen Herkunft einige der hier identifizierten Anforderungen erfüllen. Im Anschluss an die Vorstellung der vier Verfahren werden die Ergebnisse diskutiert und offene Punkte im Bereich Entwurfsunterstützung für Wearable-Systeme identifiziert.

Nr.	Name der Anforderung
A 1	Berücksichtigung der Arbeitssituation
A 2	Auswahl geeigneter Interaktionsgeräte
A 3	Entwurf angepasster Benutzungsschnittstellen
A 4	Kontextsensitivität
A 5	Abdeckung verschiedener Interaktionsgeräte
A 6	Berücksichtigung von Wearability
A 7	Berücksichtigung der Auswirkungen von Multitasking
A 8	Berücksichtigung des Non-User
A 9	Kommunikation der Arbeitssituation

Tabelle 2.1: Anforderungen an eine Entwurfsunterstützung für Wearable-Computing-Systeme.

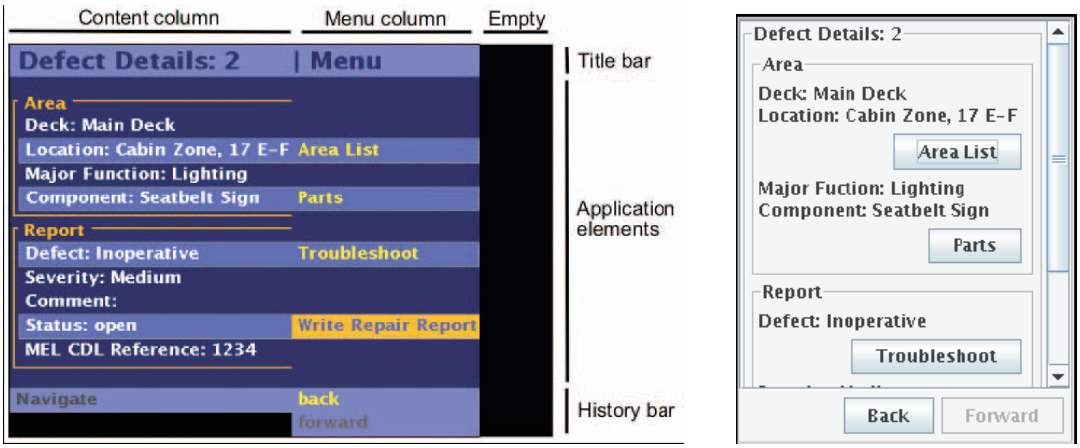

Abbildung 2.4: WUI-Toolkit Benutzungsschnittstelle für ein HMD (links) und einen PDA (rechts) ausgehend vom selben abstrakten Modell [WNK07].

2.5.1 Implementierungsunterstützung

In der Vergangenheit waren Wearable-Computing-Systeme in jeder Hinsicht Speziallösungen. Hardware, Software und auch die Interaktion mit dem Benutzer wurden individuell aufeinander abgestimmt. Derzeit ermöglicht eine zunehmende Konsolidierung der Hardware Plattform jedoch erste wiederverwendbare Architekturen zur Entwicklung von Wearable-Systemen. Diese Architekturen ermöglichen einheitliche Entwurfsprozesse und sind Voraussetzung für einen Entwurfsprozess, wie er in dieser Arbeit entwickelt wird. Um zu zeigen wie eine solche Entwicklungsunterstützung aussehen kann, wird im Folgenden das WUI-Toolkit (Wearable User Interface Toolkit) von Witt u. A. [WNK07] beschrieben.

Das WUI-Toolkit ermöglicht die abstrakte Definition der Benutzungsschnittstelle, ohne die verwendete Hardware zu kennen. Zur Laufzeit werden verschiedene Ausgabeschemata unterstützt, die eine sinnvolle Interaktion mit einer bestimmten Art von Interaktionsgerät garantieren. Das WUI-Toolkit wählt dann zur Laufzeit mit Hilfe von Regeln diejenigen verfügbaren Interaktionsgeräte aus, die am besten zu einem der Ausgabeschemata passen. Sind Kontextsensoren verfügbar, kann die Benutzungsschnittstelle automatisch an einige vordefinierte Zustände angepasst werden (z. B. Umgebungsgeräusche und Umgebungslicht).

Das WUI-Toolkit erlaubt also eine einfache Entwicklung von Wearable-Computing-Benutzungsschnittstellen und wurde bereits in mehreren Projekten erfolgreich eingesetzt [KW07, MKS+07]. Das Toolkit unterstützt den Designer jedoch nicht bei der Entscheidung welche Interaktionsgeräte oder welches Ausgabeschema in der gegebenen Arbeitssituation überhaupt sinnvoll ist.

Da es sich beim WUI-Toolkit um eine Implementierungsunterstützung handelt, sind die identifizierten Anforderungen nur eingeschränkt anwendbar. Trotzdem erfüllt es einige der Anforderungen zumindest teilweise. Die Anforderung A4 wird beispielsweise durch die eingebaute Kontextsensitivität eingeschränkt erfüllt, wenn auch kein Entwurf neuer Kontextabhängiger Funktionen vorgesehen ist. Die Anforderung A3 wird in sofern erfüllt, als dass Benutzungsschnittstellen, die mit dem Toolkit entwickelt werden, prinzipiell auf die Bedürfnisse von Wearable-Computing-

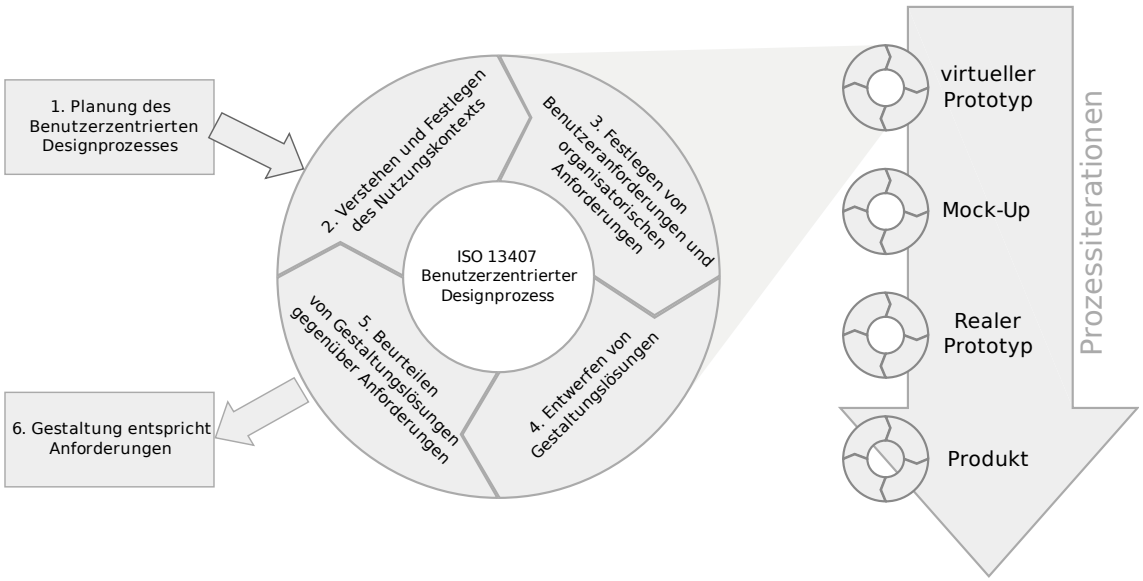

Abbildung 2.5: ISO 13407 User-centered design process [ISO99].

Benutzern angepasst sind. Allerdings deckt das Toolkit nur einige Ausgabeschema ab, was seine Verwendung stark einschränkt. Aus demselben Grund wird die Anforderung A5 ebenfalls nur teilweise erfüllt, da nur einige wenige Interaktionsgeräte unterstützt werden.

2.5.2 Entwurfsunterstützung

Um eine benutzbare Wearable-Computing-Anwendung entwerfen zu können, müssen der Benutzer und seine Arbeit im Mittelpunkt stehen. Ein Entwurfsprozess, bei dem der Benutzer im Mittelpunkt steht, wird auch als benutzerorientierter Entwurf (engl. user centered design) bezeichnet. Der in Abbildung 2.5 illustrierte ISO Standard 13407 beschreibt die wesentlichen Schritte eines solchen Prozesses [ISO99]. Um die Entwicklung von Wearable-Systemen in diesem Prozess unterstützen zu können, sind Methoden und Werkzeuge erforderlich, welche die spezifischen Aspekte des Wearable Computing berücksichtigen. Im Folgenden werden drei wesentliche Vorarbeiten vorgestellt, und zwar *Contextual Design*[BH98], das *ICE-Tool*[Bür02] sowie *Toto*[BS90]. Contextual Design ist ein allgemeiner benutzerorientierter Entwurfsprozess, der sich in vielen Anwendungsgebiete einsetzen lässt. Das ICE-Tool von Bürgy hingegen unterstützt speziell den Entwurf von Wearable-Systemen, deckt jedoch nur einen Teil der Prozesskette ab. Das Werkzeug Toto von Bleser u.A. wiederum unterstützt den Entwurf von Interaktionsmethoden unter Berücksichtigung der dazu notwendigen physischen Aktivitäten des Benutzers.

Contextual Design

Contextual Design von Beyer und Holtzblatt [BH98] ist ein benutzerorientierter Prozess, der den Entwurf von benutzbaren Anwendungen mithilfe strukturierter Vorgehensweisen erleichtert. Der beschriebene Prozess beginnt mit der Datenerhebung

beim Benutzer und schließt neben dem Entwurf von Anwendungen auch das Testen von Prototypen ein. Dabei stehen die Ziele des Benutzers stets im Vordergrund.

Der Prozess beginnt damit die Arbeit des Benutzers zu verstehen. In diesem Schritt wird ein Verfahren vorgeschlagen, das als Befragung im Kontext (engl. contextual inquiry) bezeichnet wird. Dabei wird der Benutzer während seiner normalen Arbeit zu konkreten Arbeitsschritten, den Hintergründen und der physischen Umgebung befragt. Der Befragte nimmt dabei die Rolle eines Meisters ein, während der Befrager sich wie ein Lehrling verhält.

Im nächsten Schritt werden die gewonnenen Informationen mithilfe grafischer Sprachen modelliert und die Ergebnisse, die von mehreren Benutzern gewonnen wurden in einem einzigen Modell verdichtet. Nach Aussage der Autoren deckt ein Bild Struktur und Muster einer Arbeitssituation auf, was den weiteren Entwurfsprozess erheblich vereinfacht. Gleichzeitig argumentieren sie gegen die weitverbreitete Praxis Informationen über die Arbeitssituation als Text zu dokumentieren: *„A picture is a better representation than a page of text because it's easier to see what you are talking about.“* [BH98].

Contextual Design sieht fünf Modelle vor, um eine Arbeitssituation zu beschreiben. Das *Flussmodell* beschreibt Kommunikation und Koordination zwischen Personen. Das *Sequenzmodell* listet Einzelschritte auf, die notwendig sind, um ein bestimmtes Ziel zu erreichen. Das *Artefaktmodell* beschreibt die Gegenstände, die für die Arbeit relevant sind. Das *Kulturmodell* beschreibt Politik- und Kulturbedingte Rahmenbedingungen des Arbeitsprozesses und das *Physikalische Modell* schließlich stellt die Arbeitsumgebung des Benutzers dar. Diese Modelle sind in nahezu allen Anwendungsfällen relevant, müssen jedoch nicht ausreichen. Contextual Design sieht deshalb explizit die Verwendung weiterer grafischer Modelle vor, sofern dies für eine bestimmte Arbeitssituation notwendig ist. Diese Modelle werden jedoch nicht vorgeschrieben.

Obwohl Contextual Design ein generisches Verfahren ist, sind die vorgeschlagenen Modelle und Methoden auf Büroszenarien ausgerichtet. Es wird angenommen, dass der gesamte Arbeitsprozess umgestaltet werden kann. Eine parallele Primäraufgabe wird beim Entwurf nur eingeschränkt berücksichtigt, gerade in Bezug auf Wearable Computing Anwendungen ist diese Annahme jedoch nicht richtig. Aus diesem Grund wird auch Anforderung A1 nur eingeschränkt erfüllt. Durch die flexibe Gestaltung des Entwurfsprozesses können zwar viele Szenarien abgedeckt werden, es fehlen jedoch entscheidende Informationen, die es ermöglichen würden, die gleichzeitige Durchführung von Primär- und Computeraufgabe zu verstehen und zu planen. Die existierenden Modelle haben zwar prinzipiell eine Kommunikation der Arbeitssituation (Anforderung A9) zum Ziel, können diese Aufgabe jedoch auf Grund fehlender Detailinformationen für Wearable Computing nicht vollständig erfüllen. Beispielsweise kann das beschriebene Sequenzmodell keine parallelen Aufgaben abbilden. Außerdem fehlt ein Benutzermodell, das die Fähigkeiten sowie zur Verfügung stehenden Interaktionsmöglichkeiten beschreibt. Da Contextual Design eine reine Prozessbeschreibung ist, werden außerdem keine Werkzeuge zur Verfügung gestellt, welche die Durchführung unterstützen.

ICE-Tool

Eine der wenigen Arbeiten, die sich mit der Entwurfsunterstützung für Wearable-Computing-Systeme beschäftigt hat, ist die Dissertation von Bürgy [Bür02]. Bürgy beschreibt ein „mobile and wearable computer aided engineering system“ kurz m/w-CAE System, das den Entwurf einer mobilen oder wearable Lösung für eine bestimmte Arbeitssituation unterstützt. Damit fällt seine Arbeit unter die zweite Schule der Wearable Computing Verwendungsmuster, die der spezifischen Problemlösung.

Bürgy sieht eine wachsende Unterstützung bei der Entwicklung von Wearable-Computing-Lösungen und nach wie vor ein Defizit bei der Unterstützung von Entwurfsentscheidungen. Er betrachtet in seiner Arbeit Szenarien, bei denen es um die Unterstützung einer bestehenden Primäraufgabe des Benutzers geht. Ein solches Szenario wird von Bürgy in mehrere Arbeitssituationen unterteilt, wobei jede Arbeitssituation eine Kombination aus Arbeitsort, Benutzerrolle und Arbeitsaufgabe ist. Diese Arbeitssituationen werden dann mithilfe eines Modells der Randbedingungen (engl. constraint model) beschrieben. Auf Basis dieser Beschreibung können nun vergleichbare bestehende Lösungen identifiziert oder neue Systeme entworfen werden, die an die Arbeitssituation angepasst sind.

Eine Arbeitssituation kann bei herkömmlichen Büroanwendungen durch den *Benutzer*, die *Anwendung* und die *Geräte* beschrieben werden. Bürgy erweitert dieses Modell um zwei Komponenten, die bei m/w-CAE Systemen zusätzlich eine Rolle spielen, die *Umgebung* und die *Aufgabe* des Anwenders.

Das *Benutzermodell* beschreibt, welche kognitiven Fähigkeiten dem Benutzer zur Interaktion mit dem Computersystem noch zur Verfügung stehen. Jede Modalität wird durch einen von drei Zuständen beschrieben: Steht zur Verfügung, steht eingeschränkt zur Verfügung oder steht nicht zur Verfügung. Das *Gerätemodell* ist analog zum Benutzermodell definiert. Ein Gerät kann jede der Modalitäten, die dem Benutzer zur Verfügung stehen verwenden, um eine Interaktion zu ermöglichen. Die Entscheidung ob eine Modalität von einem Gerät unterstützt wird, ist wiederum binär. Das *Anwendungsmodell* charakterisiert die Daten, mit denen der Benutzer im Computersystem umgehen muss. Für die zur Verfügung stehenden Datenarten ist die Entscheidung jeweils binär. Das *Umgebungsmodell* ist spezifisch für m/w-CAE Systeme und modelliert die Einflüsse der Umgebung auf den Anwender und die verwendeten Geräte. Das Modell erfasst Umgebungshelligkeit, Lautstärke, Sauberkeit und benötigte Robustheit einer Arbeitssituation in jeweils drei Stufen: niedrig, normal und hoch.

Bei der *Aufgabe* des Benutzers unterscheidet Bürgy zwischen drei Arten von Aufgaben, der *Primäraufgabe*, der *Unterstützungsaufgabe* und der *Kontrollaufgabe*. Die Primäraufgabe stellt die Arbeitsschritte des Benutzers dar, die ohne das Computersystem ablaufen. Die Unterstützungsaufgabe ist der Teil der Interaktion mit dem Wearable Computer, der produktiv ist, während Kontrollaufgaben dazu nötig sind, den Computer in einen produktiven Zustand zu überführen. Einer Arbeitssituation wird nun eine Kombination dieser drei Basistypen zugeordnet, die angibt, welche Typen in dieser Situation parallel ausgeführt werden müssen. Zusätzlich wird modelliert ob der Benutzer weitere Werkzeuge benötigt, ob er mit Menschen interagiert, ob er Zugriff auf externe Datenquellen benötigt und ob er seine volle Aufmerksamkeit für die Primäraufgabe benötigt wird. Alle diese Entscheidungen werden binär

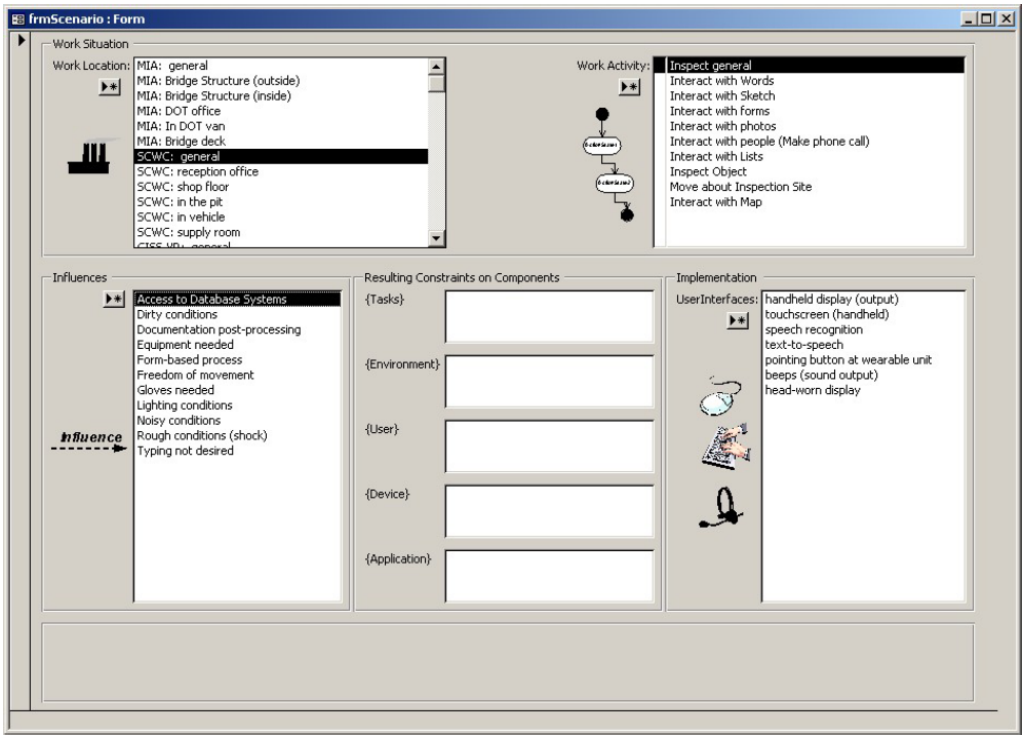

Abbildung 2.6: Screenshot des ICE-Tool Prototyps zur Modellierung von Arbeitssituationen und Gerätekonfigurationen [Bür02].

getroffen.

Mit dem in Abbildung 2.6 dargestellten ICE-Tool(Interaction Constraints Evaluation Tool) stellt Bürgy ein Werkzeug bereit, das es ermöglicht solche Modelle einer Arbeitssituation zu erstellen und anschließend in einer Datenbank nach Implementierungen zu suchen, die für eine ähnliche Arbeitssituation entwickelt wurden. Auf diese Weise kann Designwissen aus früheren Anwendungen auch durch Domänenexperten, die keine Systementwickler sind, genutzt werden. Im Rahmen der verwendeten Modellierung ist das ICE-Tool also in der Lage Szenarien abzubilden. Die starke Abstraktion der Arbeitssituationen führt jedoch dazu, dass eine Kommunikation der Arbeitssituation nur sehr eingeschränkt möglich ist (A9).

Da das Hauptziel bei der Entwicklung des ICE-Tools die Identifikation von ähnlichen Arbeitssituationen war, sind die Modelle sehr einfach gehalten und nicht geeignet um detaillierte Entwurfsentscheidungen zu treffen. Die meisten Eigenschaften der verschiedenen Modelle können nur binäre Entscheidungen festhalten. Bürgy berücksichtigt außerdem nur eine geringe Auswahl fester Modalitäten und damit auch nur eine geringe Anzahl möglicher Interaktionsgeräte. Beispielsweise kann das Modell Geräte die eine Berührung erfordern, nicht von berührungslosen Geräten unterschieden. Außerdem werden veränderte Bedingungen innerhalb einer Arbeitssituation nicht berücksichtigt. Es wird also immer die jeweils ungünstigste Situation modelliert. Kurze Zeitbereiche mit verbesserten Bedingungen können daher nicht ausgenutzt werden. Aus diesem Grund kann die Anforderung A1 zwar rudimentär erfüllt werden, detaillierte Aussagen sind hier jedoch nicht möglich. Aus demselben Grund kann auch Anforderung A2 nur teilweise erfüllt werden, da die starke Abstrak-

tion des Interaktionsgerätemodells einige sehr unterschiedliche Interaktionsgeräte als gleich erscheinen lässt und so eine differenzierte Berücksichtigung unmöglich macht.

Toto

Toto von Bleser und Siebert ist eines der wenigen Werkzeuge, die den Entwurf einer Mensch-Computer-Interaktion unter Berücksichtigung verschiedener Interaktionsgeräte und deren verwendeten Modalitäten ermöglicht [BS90]. Toto verwendet ein umfangreiches Interaktionsgerätemodell, das neben der Schnittstelle zum Computer in Form von Ein- und Ausgabebeschreibungen, auch die physikalischen Aspekte eines Interaktionsgerätes berücksichtigt. Dabei wird sowohl die physische Repräsentation des Gerätes beschrieben, als auch die Aktionen mit denen das Gerät manipuliert werden kann. Die physische Repräsentation enthält neben anderen Eigenschaften auch den Körperteil des Menschen, mit dem das Gerät bedient wird.

Mithilfe dieses Gerätemodells lassen sich spezifische Instanzen von Interkationen aus einem Baukastensystem zusammensetzen. Dabei besteht eine Aufgabe aus mehreren Operatoren, die wiederum mit verschiedenen Interaktionsgeräten umgesetzt werden können. Heuristiken versuchen dann sinnvolle Kombinationen zu ermitteln und dem Designer vorzuschlagen.

Da es sich bei Toto um ein System zum Entwurf von Schnittstellen für Desktop-Computer handelt, werden die physikalischen Aspekte jedoch nicht mit denen einer externen Arbeitssituation verknüpft, um Konflikte zu identifizieren. Toto verwendet jedoch ein einfaches Interaktionsgerätemodell, das zudem mit Handlungen des Benutzers verknüpft ist. Aus diesen Gründen erfüllt Toto die Anforderung A3 zumindest teilweise.

2.5.3 Diskussion und Ziele

Tabelle 2.2 fasst die Eignung der vier vorgestellten Vorarbeiten bezüglich der ermittelten Anforderungen zusammen. Wie deutlich zu sehen ist, werden die Wearable-spezifischen Aspekte des Entwurfs, bisher nur unzureichend von den Vorarbeiten unterstützt. Auf diesem Gebiet besteht also noch ein großer Forschungsbedarf. Insbesondere die Aspekte Wearability (A6), Multitasking (A7) und Non-User (A8) werden derzeit noch durch keine Arbeit unterstützt. Die mit Hilfe des WUI-Toolkits erstellten Benutzungsschnittstellen sind zwar begrenzt kontextsensitiv (A4), für den Entwurf neuer kontextsensitiver Funktionen wird jedoch keine Hilfestellung angeboten. Ähnlich verhält es sich mit dem Entwurf angepasster Benutzungsschnittstellen (A3). Hier lässt das WUI-Tookit außerhalb der definierten Ausgabeschemata wenig Spielraum. Toto hingegen unterstützt den Entwurf von Interaktionsmethoden mithilfe bestimmter Interaktionsgeräte, diese sind jedoch nicht auf eine bestimmte Primäraufgabe angepasst.

Wie zu erwarten, ist die Unterstützung die durch das ICE-Tool geboten wird am Umfangreichsten, da hier der Fokus auf Wearable-Computing-Systeme gelegt wurde. Aus diesem Grund deckt das ICE-Tool auch als einziges alle vorgestellten Interaktionsgeräte ab (A5). Hier besteht jedoch noch Verbesserungsbedarf, da zwar alle Interatktionsgeräte modelliert, aber nicht unterschieden werden können. Das ICE-Tool ist auch die einzige Vorarbeit, welche die Auswahl geeigneter Interaktionsgeräte

Nr.	Name	WUI-Toolkit	Contextual Design	ICE-Tool	Toto	Ziele
A 1	Arbeitssituation		☐	☐		■ - 1
A 2	Berücksichtigen Interaktionsgeräte			☐		■ - 2
A 3	Benutzungsschnittstellen	☐			☐	
A 4	Kontextsensitivität	☐				
A 5	Abdeckung Interaktionsgeräte	☐		☐		■ - 2
A 6	Tragbarkeit					☐ - 3
A 7	Multitasking					☐ - 3
A 8	Non-User					
A 9	Kommunikation der Arbeitssituation		☐	☐		■ - 1

Tabelle 2.2: Abdeckung der Anforderungen durch Vorarbeiten. Die letzte Spalte zeigt die Teilziele der vorliegenden Arbeit in Verbindung mit den Anforderungen. ☐ = teilweise/unzureichend erfüllt, ■ = erfüllt.

unterstützt (A2) und dabei die Arbeitssituation des Benutzers berücksichtigt (A1). Contextual Design berücksichtigt zwar ebenfalls die Primäraufgabe des Benutzers, aufgrund der unterschiedlichen Zielsetzung, werden wesentliche Merkmale, die für Wearable-Computing-Szenarien wichtig sind, jedoch nicht berücksichtigt. Dasselbe gilt für die Kommunikation der Arbeitssituation (A9), die zwar vorgesehen, jedoch unzureichend ist. Das ICE-Tool bietet hier, wegen der einfachen Modelle, ebenfalls nur eingeschränkte Möglichkeiten.

Auf Grund ihrer Anzahl und der unterschiedlichen zugrunde liegenden Forschungsgebiete, erschien die Bearbeitung aller dieser offenen Punkte nicht praktikabel. Deshalb wurden für die Bearbeitung in dieser Arbeit einige Punkte als Schwerpunkte herausgegriffen und zu drei Teilzielen zusammengefasst.

Teilziel 1: Entwicklung von Werkzeugen und Modellen zur Dokumentation und Kommunikation einer Wearable-Computing-Arbeitssituation.
Wie im Verlauf dieses Kapitels gezeigt wurde, bilden die Arbeitssituation des Benutzers und insbesondere dessen Primäraufgabe, die Grundlage für jede Entwurfstätigkeit. Um die Arbeitssituation berücksichtigen zu können, wie dies Anforderung A1 fordert, wird also ein Modell der Arbeitssituation benötigt. Um mit diesen Modellen zu arbeiten werden Werkzeuge benötigt. Diese Werkzeuge müssen auch in der Lage sein die Arbeitssituation zu visualisieren, sodass eine Kommunikation mit Dritten möglich wird. Dies ist nötig, um Anforderung A9 zu erfüllen.

Teilziel 2: Entwicklung von Werkzeugen und Modellen zur Unterstützung bei der Auswahl geeigneter Interaktionsgeräte
Ohne Interaktionsgeräte ist der Benutzer nicht in der Lage, seine Computeraufga-

be zu erledigen. Die Interaktionsgeräte müssen also gemäß Anforderung A2 an die Arbeitssituation des Benutzers angepasst sein. Um die Auswahl zu unterstützen ist zunächst ein Modell der Fähigkeiten und Anforderungen eines Interaktionsgerätes notwendig. Damit diese Modellierung für viele verschiedene Arbeitssituationen nützlich ist, muss sie jedoch ein breites Spektrum verschiedener Interaktionsgeräte abdecken. Dies Entspricht der Anforderung A5, da auf der Basis eines solchen Modells eine Unterstützung des Designer umgesetzt werden kann.

Teilziel 3: Berücksichtigung der Aspekte Tragbarkeit von Interaktionsgeräten und Multitasking-Situationen
Abschnitt 2.4 hat gezeigt, dass es neben der technischen Integration einer Wearable-Lösung in eine Arbeitssituation auch eine Reihe von Aspekten gibt, welche die Benutzbarkeit von Wearable-Computing-Lösungen stark beeinflussen. Diese sind Multitasking-Situationen, die Tragbarkeit von Interaktionsgeräten und der Non-User. Keiner dieser drei Aspekte wird derzeit von den verwandten Arbeiten abgedeckt, weshalb es äußerst wichtig erscheint diese Anforderungen in eine Entwurfsunterstützung zu untegrieren.

Bei der Berücksichtigung des Non-Users handelt es sich um ein überwiegend sozialwissenschaftliches Thema, das nur schwer in Systeme und Modelle zu fassen ist. Aus diesem Grund wurde hier davon abgesehen diesen Aspekt zu unterstützen. Die Aspekte des Multitasking (Anforderung A7) und der Tragbarkeit (Anforderung A6) sind dafür besser geeignet. Aus diesem Grund sollte bei den Teilzielen 1 und 2 stets darauf geachtet werden, dass diese Aspekte mit berücksichtigt werden können.

Zwei der genannten Anforderungen werden durch die drei Teilziele zunächst nicht abgedeckt. Dies sind die *Kontextsensitivität* und der *Entwurf angepasster Benutzungsschnittstellen*. Dies hat den Grund, dass beide Anforderungen einen größeren Entwicklungsanteil haben, diese Arbeit sich jedoch nur mit dem Entwurf von Wearable-Computing-Lösungen beschäftigt. Außerdem wird durch andere Forschungsgruppen bereits an beiden Anforderungen gearbeitet, weshalb der Bedarf hier nicht so hoch ist, wie bei den anderen Anforderungen. Hartmann u.A. arbeiten beispielsweise an sogenannten proaktiven Benutzungsschnittstellen, welche die hier geforderte Kontextsensitivität umsetzen [HSM08, SHF+07]. Im Bereich der angepassten Wearable-Computing-Benutzungsschnittstellen liefert das vorgestellte WUI-Toolkit erste Ansätze. Kombiniert mit anderen Ansätzen zur automatischen Generierung von Benutzungsschnittstellen, wie z. B. SUPPLE von Gajos und Weld, könnte dies bereits zu brauchbaren Ergebnissen führen. Aus diesen Gründen wurde die Bearbeitung dieser Anforderungen in der vorliegenden Arbeit nicht als essenziell angesehen.

2.6 Zusammenfassung

In diesem Kapitel wurden zunächst verschiedene Definition des Begriffs Wearable Computing erläutert und diskutiert. Anschließend wurde daraus eine Arbeitsdefinition des Begriffs Wearable Computing sowie einiger weiterer zentraler Begriffe entwickelt. Die vorgestellten Szenarien verdeutlichen, wie Wearable-Computing-Lösun-

gen in der Praxis aussehen können. Betrachtet man dazu die Vielfalt der existierenden Geräte zur Umsetzung einer Wearable-Lösung wird schnell klar, dass hier im Vergleich zu Büroanwendungen, eine deutliche Komplexitätssteigerung vorliegt. Diese Beobachtung wird zusätzlich durch die Beschreibung einiger spezifischer Probleme verdeutlicht, die beim Entwurf von Wearable-Computing-Lösungen auftreten. Aus diesen spezifischen Problemen sowie der eingangs vorgestellten Wearable-Computing-Definition, wurden Anforderungen an eine Entwurfsunterstützung für Wearable-Computing-Systeme extrahiert. Anschließend wurden Existierende Prozesse und Werkzeuge vorgestellt und es wurde gezeigt, dass diese derzeit nur einige wenige der Anforderungen erfüllen.

Aus diesen Beobachtungen leiten sich auch die Teilziele dieser Dissertation ab, den Entwurfsprozess für Wearable Computing Systeme methodisch und mit Werkzeugen zu unterstützen. Dabei sollen insbesondere die spezifischen Aspekte von Wearable-Systemen berücksichtigt werden, definiert durch die zuvor extrahierten Anforderungen. Im Weiteren werden nun zunächst die Konzepte beschrieben, mit dem diese Teilziele erreicht werden sollen, bevor auf die Details der entwickelten Methoden und Werkzeuge eingegangen wird.

Kapitel 3

Konzeption

Im vorherigen Kapitel wurde das Gebiet des Wearable Computing und dessen spezifische Entwurfsproblematiken aufgearbeitet. Nachfolgend soll nun eine Methode beschrieben werden, die den Entwurf von benutzbaren Wearable-Computing-Systemen, bezüglich einiger der im letzten Kapitel identifizierten Anforderungen unterstützt. Die Methode besteht aus drei Modellen und einem darauf aufsetzenden Entwurfsprozess.

Die Modelle beschreiben den Benutzer, die Arbeitssituation und das Computersystem sowie deren Beziehungen. Das *Arbeitssituationsmodell* bildet die Arbeitssituation ab, für die ein Wearable-Computing-System entworfen werden soll. Das *Benutzermodell* beschreibt die Interaktionsfähigkeiten des Benutzers und ermöglicht so die Simulation einer Arbeitssituation unter Berücksichtigung verschiedener Interaktionsgeräte. Das *Computersystemmodell* schließlich beschreibt die technischen Komponenten eines Wearable-Computing-Systems. Modelle konkreter Komponenten, wie z. B. von Interaktionsgeräten, werden außerdem in einer Bibliothek vorgehalten und repräsentieren bestehendes Designwissen, das dem Designer den Entwurf erleichtert. Die Dreiteilung des verwendeten Modells hat die Wiederverwendbarkeit der einzelnen Teile zum Ziel. Einzig das Arbeitssituationsmodell ist abhängig vom bearbeiteten Szenario. Das Computersystemmodell hingegen kann in verschiedenen Szenarien verwendet werden. Beide verwenden ein gemeinsames Modell des Benutzers, weshalb dieses ebenfalls eigenständig ist.

Der durch diese Modelle unterstützte benutzerorientierte Entwurfsprozess basiert auf dem in Abschnit 2.5.2 bereits erwähnten ISO Standard 13407 [ISO99], der vier zentrale Prozessschritte für den benutzerorientierten Entwurf definiert. Durch die starke Integration von Entwurf und Beurteilung werden diese beiden Prozessschritte jedoch zusammengefasst. Der hier beschriebene Prozess gliedert sich daher in die folgenden drei Abschnitte:

1. Datenerhebung
2. Modellierung
3. Entwurf

Im Rahmen der *Datenerhebung* werden verschiedene Methoden beschrieben, die sich eignen um Rohdaten für die Modellierung der Arbeitssituation zu sammeln.

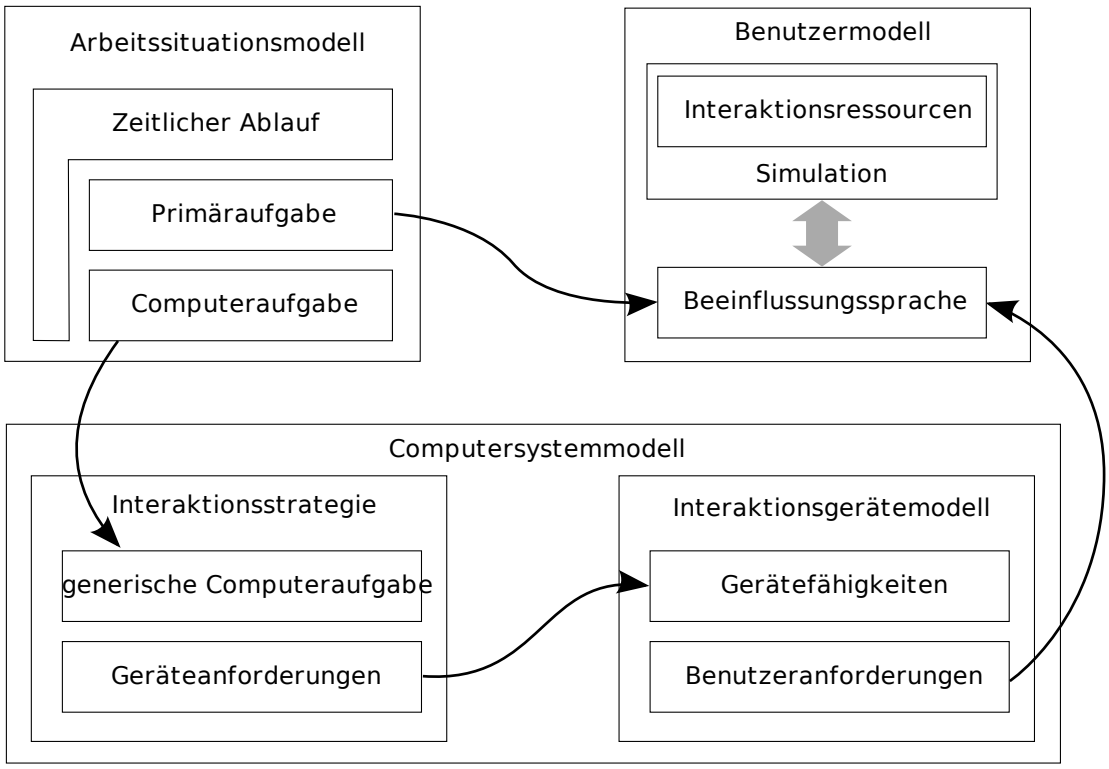

Abbildung 3.1: Komponenten des Gesamtmodells.

Zur Unterstützung wurde ein Werkzeug zur effizienteren Benutzerbeobachtung entwickelt und analysiert. Der Abschnitt *Modellierung* behandelt die Überführung der Rohdaten in ein Arbeitssituationsmodell. Im letzten Schritt, dem *Entwurf*, wird der Designer durch die drei Modelle bei Designentscheidungen und der Auswahl geeigneter Interaktionsgeräte unterstützt. Im Folgenden werden zunächst die entwickelten Modelle erläutert. Anschließend werden die drei Prozessschritte in der Reihenfolge ihrer Durchführung genauer beschrieben.

3.1 Modelle

Wie in Abschnitt 2.1 erläutert wird ein Wearable Computing System nach der hier verwendeten Definition stets für die Unterstützung einer konkreten Arbeitssituation entwickelt. Da die Computeraufgabe des Benutzers künftig parallel zu seiner Primäraufgabe durchgeführt werden soll, müssen sich beide Aufgaben die Ressourcen des Benutzers teilen. Aus dieser Beobachtung folgt die in Abbildung 3.1 dargestellte Struktur der drei Modelle. Das *Arbeitssituationsmodell* bildet die Bedingungen ab unter denen die Bedienung des Wearable-Computing-Systems stattfinden soll. Es beschreibt die *Primäraufgabe* des Benutzers, seine *Computeraufgabe* und den *zeitlichen Ablauf* der Arbeitssituation. Das *Computersystemmodell* beschreibt *Interaktionsgeräte* und *Interaktionsstrategien*, die zu einem Wearable-Computing-System kombiniert werden können. Das *Benutzermodell* schließlich beschreibt die Ressourcen, die dem Benutzer zur Interaktion mit seiner Umgebung zur Verfügung stehen.

Diese Dreiteilung findet sich auch in der Wearable-Computing-Definition von Steve Mann wieder, die Interaktionen zwischen dem Menschen (Benutzermodell), dem Wearable Computer (Computersystemmodell) und der Umgebung (Arbeitssituationsmodell) beschreibt (siehe 2.1.1). Zusammen erlauben diese drei Komponenten die *Simulation* einer geplanten Wearable-Computing-Lösung, unter Berücksichtigung der Primäraufgabe und den begrenzten Interaktionsmöglichkeiten des Benutzers. Auf diese Weise können einige Aspeke der Benutzbarkeit einer Lösung abgeschätzt werden ohne einen Prototypen entwickeln zu müssen.

Im Folgenden wird die Struktur der drei Modelle näher erläutert. Eine ausführliche Beschreibung der einzelnen Modellelemente folgt dann in Kapitel 4.

3.1.1 Benutzer

Das Benutzermodell modelliert die Ressourcen des Benutzers, die ihm die Interaktion mit einem Computersystem und seiner Arbeitssituation ermöglichen.

Definition 5 (Interaktionsressource)

Eine **Interaktionsressource** *ist eine Ressource des Benutzers, die ihm die Interaktion mit seiner Umgebung direkt oder indirekt ermöglicht.*

Beispiel 1

Die kognitiven Ressourcen des Benutzers wie Hören und Sehen ermöglichen direkt die Verwendung von Interaktionsgeräten. Aber auch der Platz am Körper selbst kann als Interaktionsressource angesehen werden, da viele Interaktionsgeräte am Körper befestigt werden müssen und der Platz dort begrenzt ist. Der Platz am Körper ermöglicht also indirekt die Interaktion.

Das Benutzermodell wird sowohl vom Arbeitssituationsmodell als auch vom Computersystemmodell referenziert, um dessen Einfluss auf den Benutzer zu beschreiben und auf diesem Weg Konflikte zwischen Computeraufgabe und Primäraufgabe identifizieren zu können.

Bei der Entwicklung des Benutzermodells standen zwei Anforderungen im Gegensatz. Einerseits sollte das Modell möglichst umfassend sein, um eine große Bandbreite an Interaktionsgeräten und Primäraufgaben modellieren zu können. Andererseits durfte das Modell aber auch nicht zu komplex werden, um den Modellierungsaufwand moderat zu halten.

Bei der Entwicklung eines solchen Modells können zwei verschiedene Ansätze verfolgt werden:

1. Ausgehend vom Menschen werden alle Aspekte des menschlichen Körpers modelliert, die prinzipiell zur Interaktion mit der Umgebung genutzt werden können.

2. Ausgehend von konkreten Szenarien und Interaktionsgeräten werden nur genau die Interaktionsressourcen modelliert, die für die untersuchten Szenarien und Geräte von Bedeutung sind.

Der erste Ansatz führt zu einem Modell, das theoretisch in der Lage ist alle existierenden und zukünftigen Interaktionsgeräte zu beschreiben. Gleichzeitig wird das Modell jedoch sehr komplex, da beispielsweise alle Muskeln (>600) des Menschen separat modelliert werden müssen. Der zweite Ansatz führt zu einem deutlich schlankeren Modell, das jedoch nicht vollständig ist. Interaktionsmethoden, die über die Szenarien und Geräte hinaus gehen, die zur Entwicklung des Modells herangezogen wurden, können daher nicht unbedingt beschrieben werden. Ein solches Modell muss aus diesem Grund leicht erweiterbar sein, um auch in Zukunft nützlich sein zu können.

Für die Entwicklung des hier vorgestellten Benutzermodells wurde der zweite Ansatz gewählt. Für diese Entscheidung gibt es vornehmlich zwei Gründe: Erstens soll der Designer eines Wearable-Systems mit möglichst wenig zusätzlichem Aufwand bei seiner Entwurfsarbeit unterstützt werden. Daher ist ein niedriger Modellierungsaufwand erwünscht. Zweitens ist die Integration einer neuen Interaktionsressource ein einmaliger Aufwand, da diese anschließend vielen Designern zur Beschreibung von Primäraufgaben und Interaktionsgeräten zur Verfügung steht.

Das Benutzermodell wurde auf der Basis der in den Abschnitten 2.2 und 2.3 beschriebenen Geräte und Szenarien entwickelt. Die Geräte wurden daraufhin untersucht welche Interaktionsressourcen benötigt werden, um sie zu bedienen. Die Primäraufgaben der betrachteten Szenarien wurden anschließend derselben Analyse unterzogen. Außerdem wurden die Szenarien auf Merkmale hin untersucht, welche die Verwendung der Geräte beeinflussen. Als Ergebnis dieser Analyse wurden drei relevante Klassen von Interaktionsressourcen identifiziert. Erstens die *kognitiven Fähigkeiten* des Benutzers wie z. B. Hören und Fühlen, im Folgenden auch als *Sensoren* bezeichnet. Zweitens *motorische Fähigkeiten*, also z. B. Sprache oder die Bewegung von Gliedmaßen.Diese werden im Folgenden als *Aktoren* bezeichnet. Drittens *Körperbereiche*, die zum Tragen von Kleidung zur Verfügung steht.

Auf Basis dieser Interaktionsressourcen ist eine Simulation der verfügbaren Interaktionsressourcen eines Benutzers möglich. Diese Simulation berücksichtigt zusätzlich Anforderungen (engl. requirements), welche die aus den Szenarien extrahierten Merkmale abbilden. Diese Anforderungen können die verfügbaren Interaktionsressourcen des Benutzers weiter einschränken. Beispielsweise wurde eine Anforderung identifiziert, die verhindert, dass der Benutzer mit den Händen bestimmte Gegenstände berührt, wie dies aufgrund von Hygienebestimmungen in Krankenhäusern der Fall sein kann (siehe Abschnitt 4.2).

Um den Einfluss von Primäraufgaben und Interaktionsgeräten auf das Benutzermodell beschreiben zu können, wird in Abschnitt 4.2.2 zusätzlich eine maschinenlesbare Beschreibungssprache definiert, die dann die Simulation einer Arbeitssituation unter Berücksichtigung des Benutzermodells ermöglicht.

3.1.2 Arbeitssituation

Das Arbeitssituationsmodell beschreibt die Umstände und die Aufgaben, die mit Hilfe eines Computersystems unterstützt werden sollen. Es besteht, wie in Abbildung 3.1 illustriert, aus den drei Teilmodellen *Primäraufgabe*, *Computeraufgabe* und *zeitlichem Ablauf*. Das Teilmodell *Primäraufgabe* beschreibt die physischen Aspekte der Arbeitssituation, also alle Einflüsse auf den Benutzer, die nicht aus der Compu-

teraufgabe resultieren, sondern aus der realen Welt. Die Primäraufgabe ist der Teil der Arbeitssituation in die das Wearable-Computing-System integriert werden muss. Das Teilmodell der *Computeraufgabe* beschreibt die Aufgaben, die der Benutzer in Zukunft mit dem Wearable-Computing-System erledigen soll. Schließlich verknüpft das Modell des *zeitlichen Ablaufs* die beiden anderen Teilmodelle und bringt diese in einen zeitlichen Zusammenhang um automatische Analysen zu ermöglichen.

Diese Verknüpfung ermöglicht die detaillierte Analyse unterschiedlicher Abschnitte einer Arbeitssituation. Probleme und Konflikte zwischen Computersystem und Primäraufgabe können so nicht nur global sondern auch feingranular identifiziert werden. Im Gegensatz zu anderen Verfahren (z. B. [Bür02]) wird also nicht für den ungünstigsten Zeitpunkt in einer Arbeitssituation entworfen, sondern für den Zeitpunkt, an dem eine Interaktion tatsächlich stattfinden muss.

Primäraufgabe

Das Modell der Primäraufgabe beschreibt den Einfluss der Arbeitssituation auf die Interaktionsressourcen des Benutzers. Es ist notwendig, um später die Kompatibilität eines Computersystems mit der Primäraufgabe evaluieren zu können. Die Primäraufgabe besteht dabei aus den folgenden Elementen:

1. Aktivitäten des Benutzers
2. Umgebungseinflüsse

Aktivitäten des Benutzers sind Arbeitsschritte, die typischerweise während der betrachteten Arbeitssituation durchgeführt werden. Beispiele aus dem Endoskopieszenario sind die Aktivitäten „Injektion vorbereiten“ und „Mit Assistenz reden“. *Umgebungseinflüsse* beeinträchtigen die Interaktionsressourcen des Benutzers ohne direkt mit einer seiner Aktivitäten zusammenzuhängen. Beispielsweise kann eine sehr laute Arbeitsumgebung eine gleichzeitige Interaktion über Audio behindern.

Um später Konflikte zwischen der Primäraufgabe und der Benutzung des Computersystems identifizieren zu können, wird der Interaktionsressourcenbedarf der verschiedenen Elemente der Primäraufgabe modelliert. Sowohl Aktivitäten als auch Umgebungseinflüsse werden auf dieselbe Weise modelliert. Beide verwenden das Benutzermodell um ihren spezifischen Einfluss auf den Benutzer in Form eines *Ressourcenprofils* zu beschreiben.

Definition 6 (Ressourcenprofil)

Ein **Ressourcenprofil** *beschreibt den Bedarf an Interaktionsressourcen, den eine einzelne Aktivität, ein Umgebungseinfluss oder ein Interaktionsgerät haben können. Ein* **Ressourcenprofil** *ist eine Kombination aus benötigten Interaktionsressourcen und Anforderungen.*

Im Falle einer Aktivität werden die Interaktionsressourcen aufgelistet, die für die Durchführung der Aktivität notwendig sind. Bei Umgebungseinflüssen werden hingegen die Interaktionsressourcen aufgelistet, deren Benutzung durch den Umgebungseinfluss behindert wird. Ressourcenprofile werden mit der *Beeinflussungssprache* spezifiziert, die in Abschnitt 4.2.2 beschrieben wird.

Computeraufgabe

Ziel des Modells der Computeraufgabe ist die Beschreibung derjenigen Teilaufgaben des Benutzers, die er in Zukunft mit dem Wearable-System durchführen soll. Um später das Zusammenspiel zwischen Computeraufgabe, Primäraufgabe und Computersystem simulieren zu können, müssen zwei Anforderungen erfüllt werden:

1. Die Computeraufgabe muss so allgemein beschrieben werden, dass sie sinnvoll mit der Primäraufgabe in einen zeitlichen Zusammenhang gebracht werden kann.

2. Die Computeraufgabe muss so detailliert beschrieben werden, dass konkrete Interaktionsgeräte zu ihrer Umsetzung ermittelt werden können.

Auch beim Entwurf von Benutzungsschnittstellen, die ohne parallele Primäraufgabe verwendet werden, wird die Computeraufgabe spezifiziert, um den Entwurfsprozess zu vereinfachen. In den letzten Jahren hat in diesem Bereich der modellbasierte Entwurf von Benutzungsschnittstellen zunehmend an Bedeutung gewonnen. Deshalb lohnt es sich, diese Methoden hier ebenfalls zu betrachten. UsiXML [LV04] und das Metamodell des EMODE-Projekts [EMO06] sind Beispiele derartiger Modellierungen, die nicht nur die Modellierung herkömmlicher, sondern auch die Modellierung multimodaler Benutzungsschnittstellen ermöglichen. Beide Methoden verwenden eine Hierarchie von Teilaufgaben, um eine Anwendung bzw. Computeraufgabe zu beschreiben. Die Blätter dieser Hierarchie sind sogenannte Interaktoren, während die höheren Hierarchieebenen der Strukturierung dienen. Interaktoren repräsentieren einzelne Interaktionen des Benutzers mit der Anwendung und können direkt mit Interaktionsgeräten implementiert werden. Elemente der höheren Hierarchieebenen erlauben hingegen das Zusammenfassen mehrerer solcher konkreter Aufgaben zu abstrakten logischen Aufgaben.

Diese Unterteilung in konkrete Teilaufgaben und übergeordnete logische Aufgaben ist geeignet, um die beiden oben genannten Anforderungen zu erfüllen. Interaktoren sind geeignet um direkt mit Interaktionsgeräten in Verbindung gebracht zu werden, während logische Aufgaben allgemein genug sind um eine sinnvolle Zuordnung zur Primäraufgabe zu ermöglichen. Da die Begriffe Interaktor und logische Aufgabe in der Praxis unterschiedlich gebraucht werden, definiert diese Arbeit analog zu den beiden Konzepten, ein Modell bestehend aus *Mikro-Computeraufgaben* und *Makro-Computeraufgaben*. Mikro-Computeraufgaben sind jedoch zunächst anwendungsspezifisch, damit der Designer später noch den Bezug zur Arbeitssituation herstellen kann. Aus diesem Grund werden zusätzlich *generische Computeraufgaben* eingeführt, die im wesentlichen einer Mikro-Computeraufgabe entsprechen, jedoch anwendungsunabhängig sind. Diese generischen Computeraufgaben sind notwendig um das existierende Designwissen der Computeraufgabe in Form einer Bibliothek abspeichern zu können.

Definition 7 (Mikro-Computeraufgabe)

Eine **Mikro-Computeraufgabe** *definiert eine elementare Interaktion mit dem Benutzer. Elementar bedeutet hier, dass die Mikro-Computeraufgabe im Kontext der Anwendung nicht sinnvoll weiter unterteilt werden kann. Eine Mikro-Computeraufgabe ist anwendungsspezifisch.*

Definition 8 (generische Computeraufgabe)

Eine **generische Computeraufgabe** *entspricht der Granularität einer Mikro-Computeraufgabe, ist im Gegensatz zu dieser jedoch anwendungsunabhängig und kann in anderen Szenarien wiederverwendet werden.*

Definition 9 (Makro-Computeraufgabe)

Eine **Makro-Computeraufgabe** *ist eine Menge von Mikro-Computeraufgaben, die aufgrund der Arbeitssituation logisch und zeitlich zusammenhängend ist. Die Granularität einer Makro-Computeraufgabe ist also so angelegt, dass die enthaltenen Mikro-Computeraufgaben in der Regel gemeinsam bzw. zeitlich nah bearbeitet werden.*

Diese zwei Ebenen reichen aus, um die Kompatibilität zwischen einer, mit einem Computersystem implementierten Computeraufgabe, und der Primäraufgabe des Benutzers zu bestimmen, wie die hier beschriebenen Beispiele und die Fallstudien in Kapitel 5 zeigen. Da das hier verwendete Modell an das modellbasierter Entwicklungsmethodiken angelehnt ist, können die verwendeten Mikro- und Makro-Computeraufgaben später leicht in ein solches Modell übertragen werden. Dadurch wird der reibungslose Übergang in einen modellbasierten Entwicklungsprozess möglich.

Die folgende Aufstellung zeigt beispielhaft, wie eine Aufteilung des Endoskopieszenarios in Mikro- und Makro-Computeraufgaben aussehen könnte. Namen in Klammern bezeichnen die zugehörige generische Computeraufgabe:

1. **Makro:** Patientenakte Einsehen
   - **Mikro:** Dokument auswählen [Auswahl aus einer Liste]
   - **Mikro:** Dokument ansehen [Navigation]
   - **Mikro:** Schließen des aktuellen Dokuments [Aktion]
2. **Makro:** Untersuchungsbegleitende Dokumentation
   - **Mikro:** Kommentar aufzeichnen [Sprachaufzeichnung]
   - **Mikro:** Aktuelle Aktivität auswählen [Auswahl aus einer Liste]
3. **Makro:** Befunddokumentation
   - **Mikro:** Befund aufzeichnen [Sprachaufzeichnung]
   - **Mikro:** Spulen in Befundaufzeichnung [Navigation]
   - **Mikro:** Untersuchungsdokumentation [Formular ausfüllen]

Makro-Computeraufgaben wie „Patientenakte Einsehen“ spezifizieren Benutzerziele, ohne auf deren konkrete Umsetzung einzugehen. Mikro-Computeraufgaben wie „Dokument auswählen“ und „Dokument ansehen“ geben dann konkrete domänenspezifische Arbeitsschritte an. Die generischen Computeraufgaben abstrahieren schließlich von der domänenspezifischen Sicht und können daher wiederverwendet werden, wie beispielsweise die Aufgaben „Auswahl aus einer Liste“ und „Navigation“.

Zeitlicher Ablauf

Das Modell des zeitlichen Ablaufs soll die Primäraufgabe mit der Computeraufgabe in einen zeitlichen Zusammenhang bringen, um die folgenden beiden Ziele zu erreichen:

1. Das Modell soll dem Designer ermöglichen die zeitliche Struktur der Arbeitssituation schnell zu erfassen und typische Situationen zu identifizieren.

2. Die Kompatibilität eines Computersystems, mit gleichzeitig stattfindenden Elementen der Primäraufgabe, muss überprüft werden können.

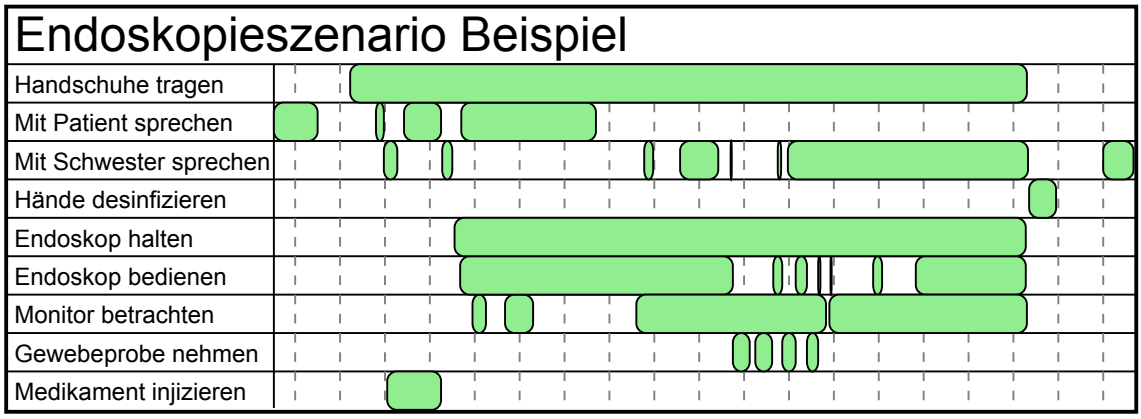

Abbildung 3.2: Ein Beispielablauf des Endoskopieszenarios. Beispiele werden durch direkte Beobachtung gewonnen. Die Zeit ist von links nach rechts abgetragen und jede Zeile repräsentiert eine Art von Ereignis.

Um diese beiden Ziele zu erreichen, wird in dieser Arbeit eine exemplarische Modellierung des zeitlichen Ablaufs in Form sogenannter *Traces* verwendet. Dabei wird eine Menge von direkt beobachteten konkreten Szenarien verwendet, um exemplarisch die zeitlichen Zusammenhänge innerhalb einer Arbeitssituation zu repräsentieren. Diese Modellierung erlaubt die in Abbildung 3.2 gezeigte intuitive Darstellung bei der wichtige Faktoren wie die Dauer von Ereignissen oder die Menge gleichzeitig auftretender Ereignisse direkt in der Visualisierung kodiert sind. Die Interpretation dieser Zeitleistendarstellung ist außerdem leicht verständlich, da kein komplexer Formalismus erlernt werden muss. Dadurch wird die Kommunikation mit Benutzern und anderen Teammitgliedern vereinfacht. Im Folgenden wird nun erläutert, warum diese exemplarische Modellierung einer vollständigen Modellierung vorgezogen wurde.

Betrachtet man existierende Software Entwicklungsprozesse, so stellt man fest, dass dort bereits Modelle für zeitliche Abläufe existieren. Die Modelle unterscheiden sich jedoch stark in Abhängigkeit von der modellierten Granularitätsstufe. Je nach Granularitätsstufe werden, wie in Abbildung 3.3 dargestellt, unterschiedliche Formalismen eingesetzt. Personenübergreifende Prozesse und Arbeitsabläufe (engl. workflows) werden beispielsweise mit formalen Modellierungssprachen wie z. B. YAWL [vdAtH05](Yet Another Workflow Language) abgebildet. Ein solches Modell

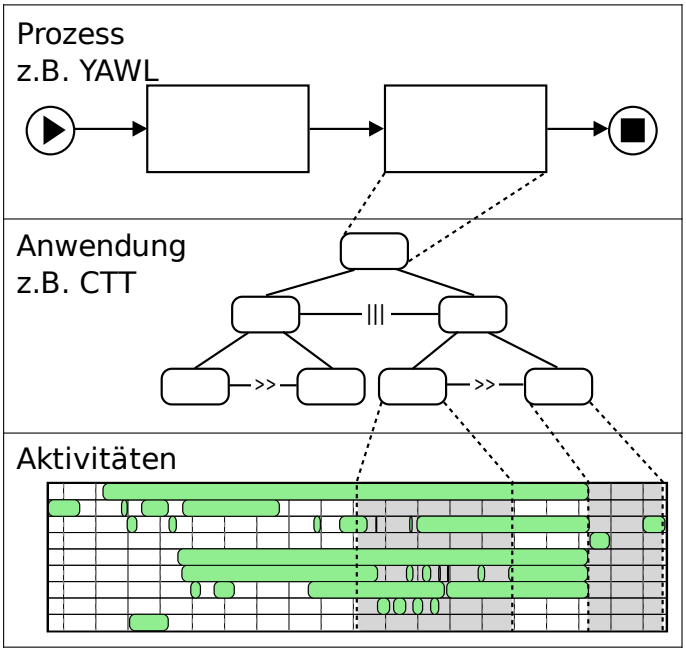

Abbildung 3.3: Aufgaben können auf verschiedenen Granularitätsstufen modelliert werden. Je nach Aufgabe werden dabei verschiedene Modellierungssprachen verwendet.

beschreibt das Zusammenspiel zwischen Systemprozessen und Benutzeraufgaben, die in der Regel in Form von Anwendungen implementiert werden. Der Ablauf innerhalb einer einzelnen Anwendung aus Sicht des Benutzers wird wiederum mit anderen Hilfsmitteln modelliert. Weit verbreitet ist die hierarchische Beschreibungssprache CTT [MPS02](ConcurTaskTrees). Sie ermöglicht die Modellierung der Benutzungsschnittstelle einer Anwendung und berücksichtigt dabei auch die Interaktion des Benutzers mit dieser Anwendung. Die Umgebung des Benutzers und seine Aktivitäten werden auf dieser Granularitätsstufe jedoch nicht berücksichtigt. Für die Modellierung der im Primäraufgabenmodell abgebildeten Aktivitäten des Benutzers wird jedoch eine noch detailliertere Granularitätsstufe benötigt.

Bei der Modellierung von Prozessen und Anwendungen werden in der Regel vollständige Modelle spezifiziert, beispielsweise mit Statecharts [Har87] oder ähnlichen Formalismen. Auf eine solche vollständige Modellierung wurde in dieser Arbeit jedoch bewusst verzichtet. Stattdessen kommt eine exemplarische Modellierung zum Einsatz, wie sie teilweise auch zur iterativen Spezifikation von Softwaresystemen verwendet wird [LMK07, UKM04].

Diese Entscheidung beruht auf verschiedenen Gründen. Zunächst kann auf der Granularitätsstufe der Aktivitäten eines Benutzers in der Regel nicht auf bestehende vollständige Modelle einer Arbeitssituation zurückgegriffen werden. Existieren solche Modelle überhaupt, dann bilden diese meist Anwendungen oder Prozesse ab, jedoch nicht die einzelnen Handgriffe, die zu deren Erledigung notwendig sind. Vollständige Modelle müssten also in den meisten Fällen zunächst erstellt werden. Dazu sind Feldstudien notwendig, um Exemplare konkreter Szenarien zu gewinnen. In einem zweiten Schritt müssten diese Exemplare dann zu einem vollständigen Modell ab-

strahiert werden. Die Verwendung eines vollständigen Modells würde also in den meisten Fällen einen zusätzlichen Arbeitsschritt bedeuten. Da die hier beschriebene Methode jedoch auch ohne vollständige Modelle angewendet werden kann, konnte dieser Arbeitsschritt eingespart werden.

Gemäß der zu Beginn dieses Abschnitts definierten Ziele müssten die vollständigen Modelle entweder die Visualisierung der Arbeitssituation oder die Qualität der Kompatibilitätsberechnung für Interaktionsgeräte verbessern. Zeitliche Aspekte wie die typische Dauer eines Ereignisses oder das gleichzeitige Auftreten mehrerer Ereignisse sind mit der hier verwendeten Zeitleistendarstellung insbesondere für die Endbenutzer intuitiv zu erkennen, da diese Informationen direkt in der grafischen Repräsentation kodiert sind. Bei einer Graphenrepräsentation (z. B. Statecharts) muss hingegen zunächst der abstrakte Formalismus erlernt werden. Das fällt Endbenutzer jedoch häufig schwer. Die Qualität der Kompatibilitätsberechnung könnte zum Beispiel dadurch verbessert werden, dass mehr typische Szenarien der betrachteten Arbeitssituation analysiert werden. Durch ein vollständiges Modell können zwar prinzipiell mehr Szenarien generiert werden, jedoch basieren diese Szenarien auf den beobachteten Exemplaren, die zur Erstellung des vollständigen Modells verwendet wurden. Deshalb können im Wesentlichen nur die Varianten abgedeckt werden, die bereits Teil der Exemplare sind. Eine Qualitätsverbesserung lässt sich hier also ebenfalls nicht erreichen..

Ein weiterer Grund für die Verwendung vollständiger Modelle, wäre die weitere Verwendung der Modelle in nachgelagerten Prozessschritten der Entwicklung. Derzeit verwenden diese nachgelagerten Prozessschritte jedoch lediglich Modelle auf den Granularitätsstufen der Prozesse und Anwendungen. Die in einem Modell der Aktivitäten enthaltenen zusätzlichen Informationen, werden also nur für die in dieser Arbeit behandelte Methode benötigt. Die bereits beschriebenen Mikro- und Makro-Computeraufgaben haben hingegen die richtige Granularitätsstufe für die Modellierung von Prozessen und Anwendungen. Die Erstellung eines vollständigen Modells der Anwendung in nachgelagerten Prozessschritten ist also direkt auf einer sinnvollen Granularitätsstufe möglich und durch die Definition der Computeraufgabe auch vorbereitet. Der Bezug zu den beobachteten Traces lässt sich durch das hier verwendete Verfahren trotzdem herstellen, sodass die Informationen über die Aktivitäten des Benutzers nicht verloren gehen. Gleichzeitig schließt die verwendete exemplarische Modellierung die Nutzung vorhandener vollständiger Modelle für die hier beschriebene Methode nicht aus. Die vollständigen Modelle lassen sich leicht dazu nutzen, eine Menge von Traces zu generieren, die dann mit den beschriebenen Verfahren verwendet werden können. Die Modellierung des zeitlichen Ablaufs mit Traces erfüllt die geforderten Ziele also ebenso gut wie vollständige Modelle, in den meisten Szenarien jedoch bei geringerem Aufwand.

3.1.3 Computersystemmodell

Das Computersystemmodell schlägt schließlich die Brücke zwischen der im Arbeitssituationsmodell definierten Computeraufgabe und dem Benutzer. Die Komponenten dieses Modells, *Interaktionsgeräte* und *Interaktionsstrategien*, bilden bereits existierendes Designwissen ab und stehen dem Designer in Bibliotheken zur Verfügung. Der Designer ist nun in der Lage diese Komponenten zum Entwurf eines vollständi-

gen Computersystems zu kombinieren, das eine bestimmte Computeraufgabe implementiert. Die zentralen Elemente dieses Modells sind die *Interaktionsgeräte*, die dem Benutzer die Interaktion mit dem Computer überhaupt erst ermöglichen. Da es vielfältige Möglichkeiten gibt, eine *Micro-Computeraufgabe* mit verschiedenen Gerätekombinationen zu implementieren, werden *Interaktionsstrategien* als Adapter zwischen Computeraufgaben und Interaktionsgeräten eingeführt. Abbildung 3.4 zeigt eine Übersicht aller im Computersystemmodell verwendeten Konzepte und deren Zusammenspiel mit den anderen beiden Modellen. Die Konzepte werden im Folgenden näher erläutert. Eine detaillierte Beschreibung des Computersystemmodells folgt später in Abschnitt 4.4.

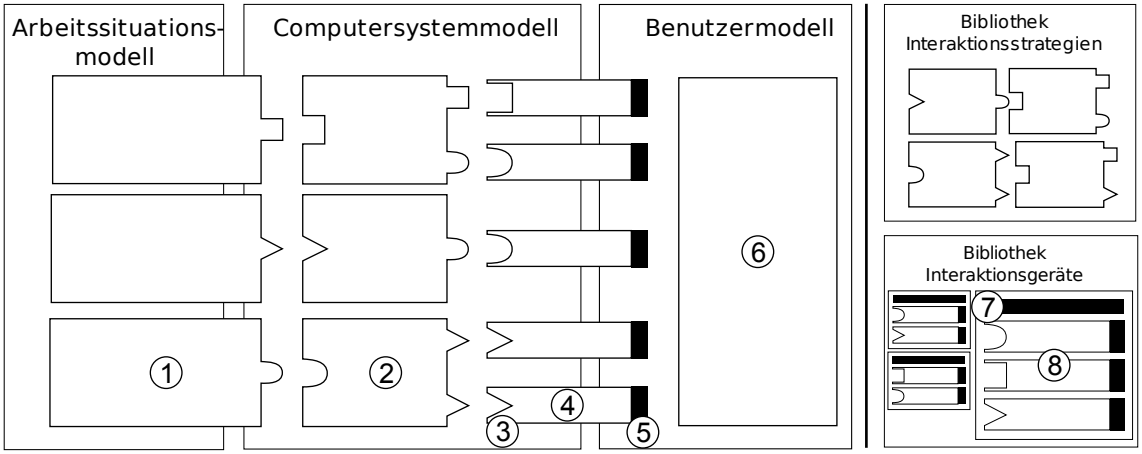

Abbildung 3.4: Komponenten des Computersystemmodells, sowie deren Zusammenspiel mit den anderen beiden Modellen: (1)generische Computeraufgabe, (2)Interaktionsstrategie, (3)Gerätefähigkeiten, (4)Datenkanal, (5)Benutzeranforderungen (aktiv), (6)Interaktionsressourcen des Benutzers, (7)Benutzeranforderungen(passiv) und (8)Interaktionsgerät.

Interaktionsgeräte

Damit der Designer feststellen kann, ob und wie eine bestimmte Mikro-Computeraufgabe mithilfe eines oder mehrerer Interaktionsgeräte implementiert werden kann, ist es notwendig die Fähigkeiten eines Interaktionsgeräts zu modellieren (*Gerätefähigkeiten*). Damit eine bestimmte Gerätekombination auch auf ihre Kompatibilität mit einer Arbeitssituation untersucht werden kann, ist es außerdem notwendig, den Interaktionsressourcenbedarf jedes Interaktionsgeräts zu modellieren (*Benutzeranforderungen*). Existierende Gerätemodelle wie das der USB-HID Spezifikation [Axe05] enthalten zwar ein sehr detailliertes Modell der Gerätefähigkeiten, Benutzeranforderungen sind jedoch nur rudimentär vorhanden. Auch das Gerätemodell von Bürgy [Bür02] enthält nur ein sehr einfaches Modell der Benutzeranforderungen. Das hier beschriebene Gerätemodell erweitert bestehende Ansätze um ein feingranulares Modell der Benutzeranforderungen, basierend auf dem bereits beschriebenen Benutzermodell. Des Weiteren werden die Benutzeranforderungen nutzungsabhängig

mit dem Modell der Gerätefähigkeiten verknüpft, um zwischen der aktiven und passiven Benutzung eines Interaktionsgerätes unterscheiden zu können.

Nachfolgend wird zunächst der Aspekt der Benutzeranforderungen betrachtet. Anschließend werden die grundlegenden Konzepte der Gerätefähigkeiten dargelegt.

Benutzeranforderungen Die Benutzeranforderungen eines Interaktionsgerätes spezifizieren die Interaktionsressourcen des Benutzers, die durch den Einsatz des Interaktionsgerätes in Anspruch genommen werden. Die Spezifikation stützt sich dabei auf das in dieser Arbeit entwickelte Benutzermodell.

Untersucht man die Interaktionsgeräte aus Abschnitt 2.2 hinsichtlich ihres Bedarfs an Interaktionsressourcen, so stellt man fest, dass dieser Bedarf über die Dauer der Benutzung hinweg nicht konstant ist. Es muss zunächst zwischen dem *aktiven* und dem *passiven Gebrauch* unterschieden werden. Der aktive Gebrauch bezeichnet dabei den Zeitraum in dem das Gerät aktiv für die Interaktion mit dem Computer verwendet wird. Viele, vor allem mobile Interaktionsgeräte benötigen jedoch auch zwischen diesen aktiven Phasen Interaktionsressourcen. Ein Datenhandschuh (z. B. [BNSS01]) beispielsweise behindert auch dann die Interaktionsfähigkeit des Benutzers, wenn dieser gerade keine Eingabe mit dem Handschuh durchführt. Bereits das Tragen des Handschuhs schränkt die Bewegungsfähigkeit der Finger ein und verhindert beispielsweise die gleichzeitige Verwendung eines Twiddlers [Sta99]. Dies wird im Folgenden als *passiver Gebrauch* eines Interaktionsgerätes bezeichnet.

Weiterhin kann ein einzelnes Interaktionsgerät unterschiedliche Arten der Interaktion vereinen, die sich wiederum in ihrem Ressourcenbedarf unterscheiden. Eine Funktion des eben genannten Datenhandschuhs ermöglicht die Interaktion durch Rotation des Handgelenks, eine weitere Funktion wird durch das Krümmen einzelner Finger aktiviert. Es ist daher notwendig den Ressourcenverbrauch nicht pro Gerät zu betrachten, sondern getrennt nach unabhängigen Funktionen.

Der aktive und passive Ressourcenbedarf eines Interaktionsgerätes, wird mit der Beeinflussungssprache des Benutzermodells modelliert. Ein Ressourcenprofil spezifiziert dabei den passiven Ressourcenbedarf des Interaktionsgerätes. Ein oder mehrere weitere Ressourcenprofile spezifizieren dann den Ressourcenbedarf der verschiedenen Gerätefähigkeiten. Einem Ressourcenprofil können dabei mehrere Funktionalitäten zugewiesen werden, sofern diese dieselben Interaktionsressourcen benötigen, aber gleichzeitig benutzt werden können.

Bei einigen Geräten sind zusätzlich Varianten in der Verwendung möglich, was ebenfalls einen Einfluss auf die benötigten Ressourcen haben kann. Je nach Position eines textilen Bedienelementes [TMTP02] müssen für die Interaktion beispielsweise unterschiedliche Körperteile bewegt werden. Solche Varianten werden im vorliegenden Modell als eigenständige Interaktionsgeräte modelliert, um die Komplexität des Modells gering zu halten.

Gerätefähigkeiten Das Modell der Gerätefähigkeiten eines Interaktionsgerätes beschreibt dessen Funktionalitäten, die einem angeschlossenen Computer zur Verfügung gestellt werden. Eine Funktionalität kann dabei entweder eine Ein- oder eine Ausgabemöglichkeit sein. Angelehnt an Arbeiten von Buxton, Lipscomb und Card [CMR91, Bux83, LP93] wird jede Funktionalität durch genau einen *Datenka-*

nal modelliert. Diese Datenkanäle bilden daher die Basis für die Beschreibung der Gerätefähigkeiten eines Interaktionsgerätes.

Definition 10 (Datenkanal)

Ein **Datenkanal** *transportiert Informationen vom Interaktionsgerät zum Benutzer oder umgekehrt. Ein Datenkanal transportiert dabei jeweils nur so viele Informationen, dass diese sinnvoll alleine interpretiert werden können.*

Beispiel 2

Die X- und die Y-Achse einer Computermaus sind beispielsweise getrennte Datenkanäle, da eine Anwendung beide Informationen separat interpretieren kann. Die Pixel eines Bildschirms hingegen ergeben nur als Ganzes einen Sinn für den Benutzer. Deshalb werden sie gemeinsam als ein einziger Datenkanal modelliert.

Jeder Datenkanal beschreibt in welcher Form die über ihn übermittelten Daten übertragen werden und welche Bedeutung diese haben. Die Auswahl der Merkmale, die für jeden Datenkanal erfasst werden, beruht ebenfalls auf den oben genannten Arbeiten zur Beschreibung von Eingabegeräten sowie auf dem HID Protokoll der USB Spezifikation [Axe05]. Details zu den verwendeten Merkmalen sowie zur Beschreibung der Datenkanäle befinden sich in Abschnitt 4.4.1. Die in diesen Arbeiten benannten Merkmale wurden zusammengefasst und punktuell um ergonomische Merkmale und Semantik erweitert. Zusätzlich werden Beziehungen zwischen verschiedenen Kanälen genauer beschrieben und stärker berücksichtigt. Das Modell wurde außerdem um eine Verknüpfung mit dem Modell der Benutzeranforderungen erweitert, um den aktiven und passiven Ressourcenbedarf zu beschreiben.

Durch diese abstrakte Beschreibung der Gerätefähigkeiten eines Interaktionsgerätes, können ähnliche Interaktionsgeräte, die zwar dieselbe Computerschnittstelle aufweisen, jedoch unterschiedliche physikalische Ausprägungen haben, aus Sicht des Computers als gleich betrachtet und so flexibler eingesetzt werden. Dazu ist jedoch auch eine anwendungsseitige Unterstützung des Interaktionsgerätemodells notwendig. Diese Unterstützung bilden hier Interaktionsstrategien, deren Konzept nun erläutert wird.

Interaktionsstrategien

Widgets sind ein weitverbreiteter Ansatz um bestimmte Aufgaben in Benutzungsschnittstellen zu kapseln, und so deren Wiederverwendung zu ermöglichen. Dabei werden konkrete Widgets jedoch für eine bestimmte Hard- und Software-Umgebung (z. B. WIMP) implementiert. Die Verwendung eines solchen Widgets ist daher nur mit genau dieser Umgebung möglich. Der Einsatz neuer Interaktionsgeräte erfordert also die Implementierung neuer Widgets. Bei der Analyse der in Kapitel 2.2 aufgelisteten Interaktionsgeräte fällt jedoch auf, dass viele Geräte ähnliche Funktionalitäten anbieten, die allerdings auf sehr unterschiedliche Weise implementiert sein können. Diese Beobachtung bestätigt sich, wenn man mit dem oben beschriebenen Modell der Gerätefähigkeiten, von der konkreten Hardware abstrahiert. Verwendet man also ein abstraktes Gerätemodell zur Beschreibung von Interaktionsgeräten, lässt sich

ein Widget unabhängig von konkreten Interaktionsgeräten implementieren, wie im Folgenden deutlich wird. Ein solches implementierungsunabhängiges Widget wird im Folgenden als *Interaktionsstrategie* bezeichnet.

Definition 11 (Interaktionsstrategie)

Eine **Interaktionsstrategie** *kapselt eine gewisse Art und Weise auf die eine generische Computeraufgabe mit einer Kombination von Datenkanälen implementiert werden kann.*

Interaktionsstrategien sind also wie in Abbildung 3.4 dargestellt, das Bindeglied zwischen der Computeraufgabe des Benutzers und den Interaktionsgeräten. Auf der einen Seite implementieren sie eine bestimmte generische Computeraufgabe, verwenden auf der anderen Seite jedoch abstrakte Datenkanäle.

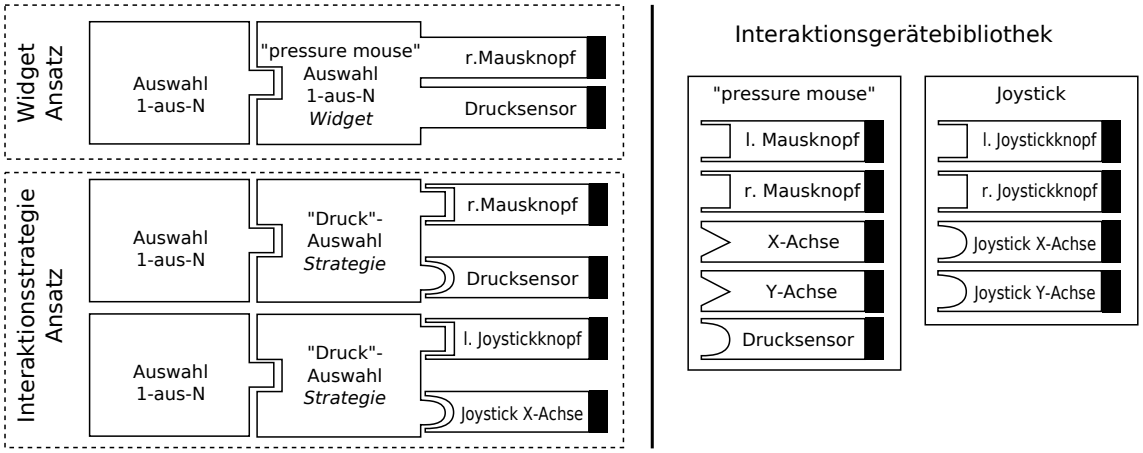

Abbildung 3.5: Das Widget „pressure mouse“ implementiert die Mikro-Computeraufgabe *Auswahl 1-aus-N* mit einer um einen Drucksensor erweiterten Maus [CIS07]. Die Interaktionsstrategie „Druck“-Auswahl abstrahiert von konkreter Hardware und lässt sich daher auch mit einem Joystick verwenden, der ähnliche Daten liefert.

Anhand eines Beispiels aus der Forschungsliteratur soll nun gezeigt werden wie das Konzept der Interaktionsstrategie genutzt werden kann, um neue Interaktionsgeräte mit geringem Aufwand, zur Implementierung einer generischen Computeraufgabe zu nutzen.

In einer Arbeit von Cechanowicz u. A. [CIS07] wurde eine Computermaus mit einem zusätzlichen Drucksensor bestückt. Die Arbeit beschreibt ein Verfahren, das die vereinfachte Auswahl eines Elements aus einer Liste mit Hilfe dieses zusätzlichen Sensors ermöglicht. Das Eingabegerät und die Interaktionsstrategie werden hierbei als Einheit beschrieben (vgl. Abb. 3.5). Wendet man die Trennung in Strategie und Gerät auf dieses Beispiel an, so erhält man zunächst ein Gerät mit verschiedenen abstrakten Datenkanälen. Der Drucksensor beispielsweise liefert einen kontinuierlichen Wert, der beim Ende der Interaktion auf ein definiertes Nulllevel zurückkehrt. Aufbauend auf dieser abstrakten Gerätebeschreibung, lässt sich nun eine Strategie formulieren, die mit dieser Art von Eingabe die generische Computeraufgabe *Aus-*

wahl 1-aus-N realisiert. Durch diese Aufteilung lässt sich die Strategie einerseits auch auf ähnliche Eingabegeräte wie Joysticks anwenden, andererseits lässt sich der Drucksensor aber auch für andere Strategien gebrauchen.

Eine Interaktionsstrategie setzt also keine bestimmten Interaktionsgeräte voraus. Es werden auf Basis des Modells der Gerätefunktionalität lediglich Anforderungen an die verwendeten Datenkanäle formuliert. Mit welchen physischen Geräten die Interaktionsstrategie später implementiert wird, spielt daher für die Strategie keine Rolle. Durch diese Abstraktion der Interaktionsgeräte wird die Flexibilität des Systems deutlich erhöht. Neue Interaktionsgeräte können bereits existierende Interaktionsstrategien nutzen, sofern das Gerät die Anforderungen der Strategie erfüllen kann. Die Abstraktion von konkreter Hardware ermöglicht außerdem, die Verteilung der Interaktion auf verschiedene Geräte. Die oben beschriebene Strategie könnte beispielsweise auch mit einer normalen Maus und einem separaten Drucksensor zusammenarbeiten.

3.2 Prozessüberblick

Aufgrund ihrer in Kapitel 2.1 beschriebenen Eigenschaften greifen Wearable-Computing-Systeme viel tiefer in den Alltag ihres Benutzers ein, als andere mobile oder stationäre Computersysteme. Aus diesem Grund spielt die Benutzbarkeit für die Akzeptanz eines solchen Systems eine noch bedeutendere Rolle. Um ein erfolgreiches Wearable-Computing-System zu entwickeln, ist es daher notwendig, den Benutzer frühzeitig und kontinuierlich in den Entwurfsprozess einzubeziehen. Dieses Vorgehen wird als benutzerorientierter Entwurfsprozess (engl. user centred design, kurz UCD) bezeichnet. Der ISO Standard 13407 mit dem Titel „Human centered design processes for interactive systems“ [ISO99] beschreibt einen prototypischen Prozess, der als Grundlage für konkrete Entwurfsprozesse dienen kann. Die sechs Prozessschritte, von denen die Schritte 2-5 die eigentliche Entwicklungsarbeit darstellen, sind in Abbildung 3.6 als innerer Ring dargestellt. Diese Schritte bilden einen Zyklus und jeder dieser Zyklen ist eine Iteration im Entwicklungsprozess. Die Entwürfe werden dabei in jeder Iteration konkreter. Beginnend mit virtuellen Prototypen am Reißbrett, werden anschließend Mock-Ups, dann reale Prototypen und schließlich das fertige Produkt entworfen und evaluiert.

Für die Entwicklung von Wearable-Computing-Systemen wurden Prozesse nach diesem Vorbild bereits in verschiedenen Projekten erfolgreich eingesetzt [ABK+08, TH04]. Aus diesem Grund ist auch der hier vorgestellte Prozess in diesen ISO-Standard eingebettet. Die drei Prozessschritte *Datenerhebung*, *Modellierung* und *Entwurf* sind in Abbildung 3.6 in Form eines äußeren Ringes dargestellt um die Parallelität mit dem ISO-Prozess zu verdeutlichen. Die ersten beiden Schritte entsprechen direkt den ISO 13407 Schritten 2 und 3. Der dritte Prozessschritt weicht allerdings in sofern von der ISO-Norm ab, als dass er Entwurf und Evaluierung gleichzeitig behandelt. Diese Eigenschaft ergibt sich aus der Vorgehensweise Entwurf und Evaluierung durch die beschriebenen Modelle direkt zu koppeln. Da die hier beschriene Methode allerdings auf virtuellen Entwurf und Evaluierung von Wearable-Computing-Systemen ausgelegt ist, kann der Hauptnutzen vor allem in den ersten Iterationen des Entwicklungsprozesses erzielt werden. Mit zunehmender Konkreti-

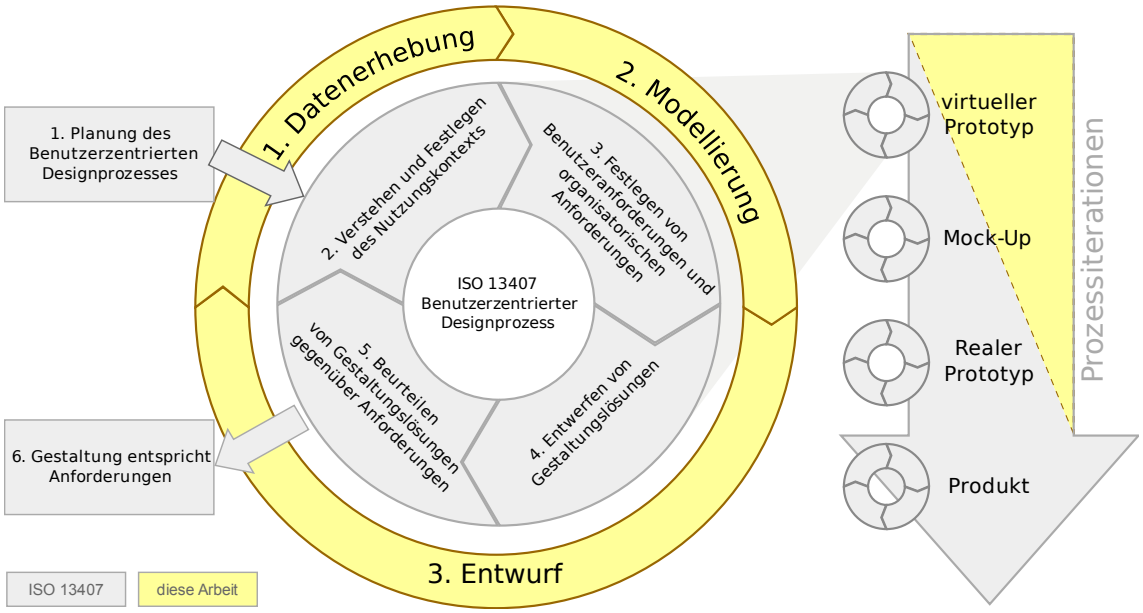

Abbildung 3.6: Die drei Phasen des durch diese Methode unterstützten Entwurfsprozesses sind nach dem ISO 13407 Prozessmodell für benutzerorientierten Entwurf angeordnet. Die drei Prozessschritte Datenerhebung, Modellierung und Entwurf stellen dabei eine Iteration im Entwurfsprozess dar. Der Hauptnutzen dieser Methode liegt jedoch in den ersten Iterationen mit virtuellen Prototypen und nimmt mit zunehmender Konkretisierung des Entwu rfs ab.

sierung der Entwürfe und steigendem Anteil des Implementierungaufwandes nimmt der Nutzen der hier vorgestellten Methode ab. Die Modelle können jedoch als Startpunkt für die weitergehende Entwicklungsarbeit verwendet werden.

Innerhalb des Prozesses werden drei wesentliche Rollen berücksichtigt: Der *Benutzer*, der *Beobachter* und der *Designer*. Der *Benutzer* steht gemäß den Prinzipien des UCD stets im Mittelpunkt aller Aktivitäten und wird regelmäßig einbezogen, um Konzepte und Ideen zu validieren. Der Entwurfsprozess selbst wird jedoch in der Regel nicht von einer einzigen Person durchgeführt, sondern von einem ganzen Team. Der hier beschriebene Teil des Entwicklungsprozesses wird im wesentlichen durch zwei Rollen bestimmt. Der *Beobachter* ist in der Regel ein Experte für Feldforschung und Benutzerstudien und ist dafür zuständig, die Arbeitssituation des Benutzers zu verstehen und die Bedürfnisse des Benutzers zu identifizieren. Der *Designer* hingegen konzentriert sich auf den Entwurf eines möglichst effizienten Computersystems, das den Anforderungen des Benutzers gerecht wird. Deshalb ist es auch notwendig die Ergebnisse, der durch den Beobachter durchgeführten Anforderungsanalyse an den Designer zu kommunizieren. Traditionell geschieht diese Kommunikation über Anforderungen (engl. Requirements) oder Berichte in Form von Fließtext. Moderne UCD-Prozesse wie Contextual Design [BH98] haben aber gezeigt, dass diese Kommunikationsformen oft nicht ausreichend sind, um den Nutzungskontext zu beschreiben.

Der hier beschriebene Prozess verwendet deshalb die beschriebenen Modelle, um spezifische Anforderungen von Wearable-Computing-Szenarien besser zu kommuni-

zieren und zu verarbeiten. Zu diesem Zweck wurden im Rahmen dieser Arbeit auch einige prototypisch implementierte Anwendungen entwickelt. Diese sind der *Task-Observer*, der den Beobachter bei seiner Arbeit unterstützt, der *WearableDesigner*, der in der Modellierung, Visualisierung und Analyse zum Einsatz kommt und der ProjectEditor, der ebenfalls die Modellierung ermöglicht. Alle Anwendungen werden in Kapitel 4 näher beschrieben.

Die folgenden drei Abschnitte beschreiben nun zunächst die einzelnen Prozessschritte und erläutern, wie die vorgestellten Modelle den Entwurfsprozess unterstützen.

3.3 Prozessschritt 1: Datenerhebung

Gemäß dem ISO 13407 Prozessmodell muss zunächst der Nutzungskontext festgestellt werden. Im Falle einer Wearable-Computing-Anwendung besteht der Nutzungskontext, neben den Anforderungen herkömmlicher Desktopsysteme, aus der Primäraufgabe und der zu implementierenden Computeraufgabe sowie deren Zusammenspiel. Diese Informationen werden durch das Modell der Arbeitssituation abgebildet. Daher müssen zunächst Informationen erhoben werden, welche die Erstellung dieses Modells ermöglichen. Nach Beyer und Holtzblatt ist dazu Feldforschung bei den zukünftigen Benutzern notwendig, da nur sie zuverlässig über ihre tägliche Arbeit berichten können [BH98]. Bei einem solchen Vor-Ort-Besuch können verschiedene Methoden wie z. B. Interviews und direkte Beobachtungen verwendet werden, um an die gewünschten Informationen zu gelangen. Jede dieser Methoden hat spezifische Stärken und Schwächen. Die Auswahl geeigneter Methoden für ein spezifisches Projekt hängt also von verschiedenen Faktoren ab.

Im Folgenden werden zunächst die *relevanten Daten* beschrieben, die für die Erstellung eines Modells der Arbeitssituation benötigt werden. Anschließend werden verschiedene *Methoden der Feldforschung* vorgestellt und untersucht: Zum einen auf ihre Stärken und Schwächen bezüglich der hier benötigten Informationen, zum anderen auf Indikatoren, die eine Verwendung der jeweiligen Methode nahelegen.

3.3.1 Relevante Daten

Um die Teilkomponenten einer Arbeitssituation modellieren zu können, müssen zunächst Rohdaten erhoben werden, aus denen dann im nächsten Prozessschritt das Arbeitssituationsmodell extrahiert wird. Um das Primäraufgabenmodell erstellen zu können, ist eine Analyse der derzeitigen *Aufgabenstruktur* des Benutzers notwendig sowie eine Untersuchung der *Umgebungseinflüsse*. Für die Modellierung der Computeraufgabe wird die Aufgabenstruktur ebenfalls benötigt. Zusätzlich sind Informationen über den *Ressourcenbedarf* von Aktivitäten und Umgebungseinflüssen zu sammeln, um den Ressourcenbedarf der Primäraufgaben definieren zu können. Die zeitlichen Abläufe werden schließlich wie in Abschnitt 3.1.2 beschrieben, mit mehreren *Traces* modelliert.

Aufgabenstruktur

Die Aufgabenstruktur des zu unterstützenden Benutzers wird benötigt, um die relevanten Teile der Primäraufgabe sowie der Comupteraufgabe zu bestimmen. Die Teilaufgaben einer Arbeitssituation können in der Regel iterativ weiter unterteilt werden, sodass eine Hierarchie von Teilaufgaben entsteht. Diese Vorgehensweise hat sich für den benutzerorientierten Entwurf von Benutzungsschnittstellen etabliert und kann als Standard angesehen werden [HR98]. Eine solche Anordnung wird als hierarchische Aufgabenanalyse (engl. hierarchical task analysis, kurz: HTA) bezeichnet [Ann04] und beschreibt die logische Aufteilung einer Arbeitssituation auf verschiedenen Granularitätsstufen.

Abbildung 3.7 zeigt beispielhaft die Aufgabenstruktur einer Endoskopieuntersuchung aus Sicht des durchführenden Arztes. Für diesen gliedert sich die gesamte Untersuchung in drei Teilaufgaben: die „Vorbereitung“, die eigentliche „Untersuchung“ und die anschließende „Dokumentation“. Alle drei Teilaufgaben können wiederum unterteilt werden, bis die gewünschte Granularitätsstufe erreicht ist. Da die Aufgabenstruktur hier im Wesentlichen zur Modellierung der Primäraufgaben verwendet wird, kann die geeignete Granularitätsstufe wie folgt beschrieben werden:

Regel 1

Die richtige Granularitätsstufe für eine Teilaufgabe *ist für den hier beschriebenen Anwendungsfall genau dann erreicht, wenn sie durch nur eine wesentliche manuelle Handlung des Benutzers bestimmt ist, die durch ein konstantes Ressourcenprofil beschrieben werden kann.*

Diese Bedingung ist notwendig um eine sinnvolle Modellierung des Ressourcenbedarfs zu ermöglichen. Ändert sich das Ressourcenprofil einer Aufgabe während ihrer Dauer, ist nur noch eine Aussage auf Basis des ungünstigsten Ressourcenprofils möglich. Eventuelle Zeitbereiche, in denen eine Interaktion dennoch möglich gewesen wäre, können so nicht mehr identifiziert werden.

Umgebungseinflüsse

Neben den eigentlichen Arbeitsschritten kann auch die Arbeitsumgebung selbst Einfluss auf den Entwurf der Computerunterstützung haben. Daher müssen diese Umgebungseinflüsse ebenfalls identifiziert und modelliert werden. Hackos und Redish identifizieren drei wesentliche Kategorien von Umgebungseinflüssen [HR98]: die physische Umgebung, soziale Aspekte und kulturelle Aspekte. Der Volere Anforderungsanalyseprozess [RR99] berücksichtigt ebenfalls nicht-funktionale Anforderungen der Kategorien: Aussehen, Benutzbarkeit und Menschlichkeit, Leistung, Betriebsumgebung, Wartung und Support, Sicherheit, kulturelle und politische Aspekte und rechtliche Rahmenbedingungen. Dies alles sind Kategorien von Anforderungen, die den Entwurf eines Computersystems beeinflussen können. Allerdings werden nicht alle diese Kategorien für die Modellierung der hier beschriebenen Arbeitssituation benötigt. Für die Modellierung der Arbeitssituation reicht es daher aus, die Anfor-

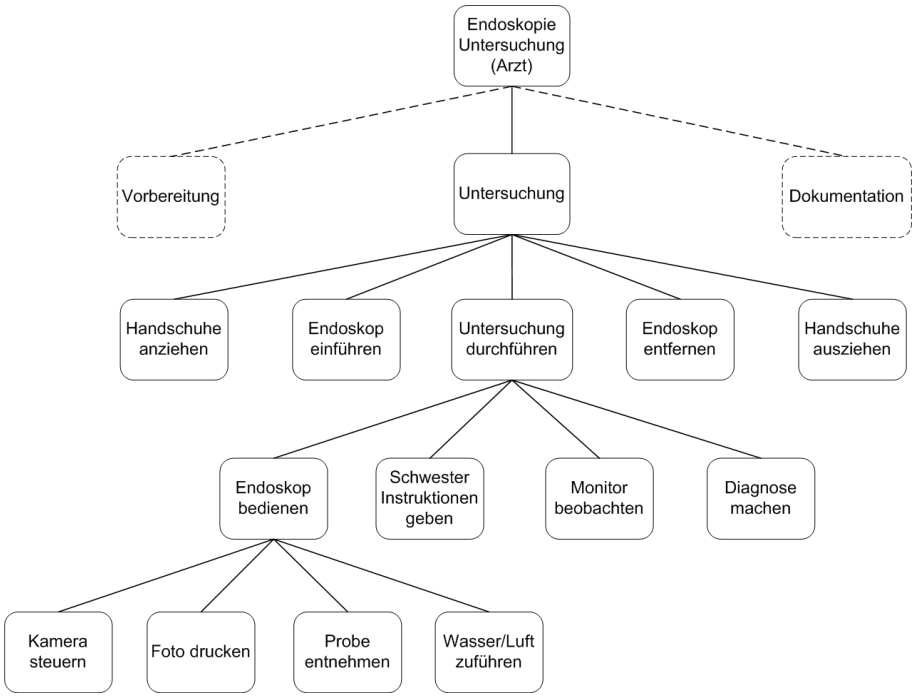

Abbildung 3.7: Hierarchische Aufgabenanalyse (HTA-Modell) des Endoskopie Beispiels aus Sicht des Arztes.

derungen zu betrachten, die folgende Regel erfüllen:

Regel 2

Für die Modellierung einer Arbeitssituation sind all diejenigen **Umgebungseinflüsse relevant**, *die zeitweise oder dauerhaft Einfluss auf die verfügbaren Interaktionsressourcen des Benutzers haben.*

Untersucht man die oben genannten Kategorien auf diese Bedingung hin und überprüft das Ergebnis mit den Erfahrungen vergangener Wearable Computing Projekte, bleiben nur einige wenige Kategorien übrig. Diese können in den folgenden Kategorien zusammengefasst werden:

Soziale und kulturelle Umgebung: Anforderungen, die sich aus der sozialen und kulturellen Umgebung des Benutzers ergeben. Der in Abschnitt 2.4.4 beschriebene Non-User ist ein wesentlicher sozialer und kultureller Aspekt bei Wearable Computung Anwendungen. **Beispiel:** *Ärzte sind nicht bereit einen Computer per Sprache zu steuern während Patienten anwesend sind, da sie Angst haben sich lächerlich zu machen.*

Physische Umgebung: Unter diese Kategorie fallen Anforderungen an das Wearable-Computing-System, die sich aus der physischen Umgebung des Benutzers ergeben. Dies schließt z. B. auch Anforderungen an die Robustheit und Schutzbeständigkeit verwendeter Geräte ein. **Beispiel:** *Eine hohe Umgebungslautstärke kann die Verwendung von Audio-basierter Interaktion verhindern.*

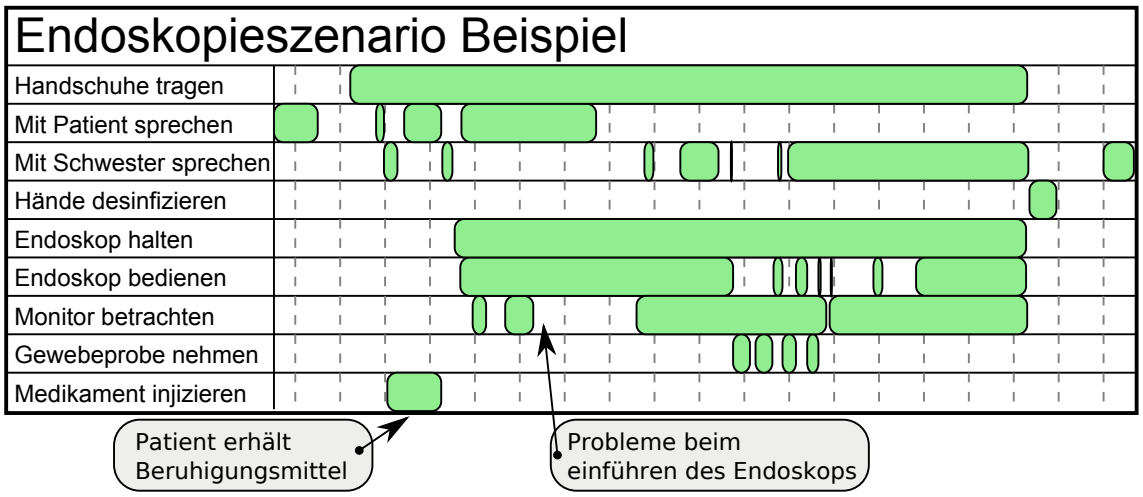

Abbildung 3.8: Beispiel eines Traces mit Anmerkungen. Es zeigt den Beginn und die zeitliche Ausdehnung eines jeden relevanten Ereignisses eines Ablaufs.

Regulatorische Rahmenbedingungen: Gesetze und interne Firmenrichtlinien können die Verwendung von bestimmten Geräten und Interaktionsformen beeinflussen. Hier spielen auch Sicherheitsaspekte eine Rolle. **Beispiel:** *Hygienevorschriften in einem Krankenhaus, schreiben in manchen Situationen Handschuhe vor und regulieren, was der Benutzer berühren darf, ohne sich desinfizieren zu müssen.*

Interaktionsressourcen / Modalitäten

Nachdem die Aufgabenstruktur der Arbeitssituation bekannt ist und auch relevante Umgebungseinflüsse identifiziert wurden, muss auch deren Ressourcenbedarf modelliert werden. Die Modellierung selber ist von den im Benutzermodell abgebildeten Ressourcen abhängig. Die Fragestellungen bei den Gesprächen mit dem Benutzer und bei Beobachtungen müssen also an das Benutzermodell angepasst werden. Für die Modellierung der benötigten kognitiven und motorischen Fähigkeiten müssen die Modalitäten identifiziert werden, die von Aktivitäten, verwendeten Geräten und Umgebungseinflüssen beeinträchtigt werden. Für die Modellierung der benötigten Körperbereiche gilt es die getragenen Kleidungsstücke und Geräte zu identifizieren sowie Umgebungseinflüsse die darauf Einfluss haben könnten.

Beispiele Zeitlicher Zusammenhänge

Das Modell des zeitlichen Ablaufs basiert auf Traces tatsächlicher Abläufe. Diese Traces müssen also ebenfalls in diesem Prozessschritt erhoben werden. Dafür müssen alle Vorkommnisse eines Elements der Primäraufgabe festgehalten werden. Abbildung 3.8 zeigt einen solchen Trace, der zusätzlich Anmerkungen des Beobachters enthält.

3.3.2 Methoden der Feldforschung

Um die beschriebenen relevanten Informationen zu erheben, stehen dem Beobachter verschiedene benutzerorientierte Methoden der Feldforschung zur Verfügung. Kuniavsky und Beyer beschreiben verschiedene Methoden der Benutzerforschung, die im Rahmen eines benutzerorientierten Entwicklungsprozesses eingesetzt werden können [Kun03, BH98]. Consolvo u. A. [CAF02] geben einen Überblick über arbeitswissenschaftliche Methoden, die im Rahmen eines Ubiquitous Computing Projektes sinnvoll angewendet wurden. Von diesen Methoden sind jedoch nur einige für die Erhebung der hier benötigten Informationen geeignet. Diese Methoden können in drei Gruppen zusammengefasst werden, die im Folgenden beschrieben werden:

- Interview
- Befragung im Kontext
- Sequenzanalyse

Die Methoden dieser Kategorien liefern teilweise redundante Daten, unterscheiden sich jedoch in puncto Qualität und benötigtem Aufwand. Um die Auswahl geeigneter Methoden für ein konkretes Szenario zu unterstützen, werden die Methoden deshalb außerdem hinsichtlich folgender Kriterien untersucht:

- Zeitaufwand der Nachbereitung
- Eignung für...
  - ... Aufgabenstruktur
  - ... Umgebungseinflüsse
  - ... Interaktionsressourcen
  - ... Beispiele zeitlicher Zusammenhänge
- Technische/organisatorische Voraussetzungen

Der benötigte Aufwand einer Methode ist deshalb von Bedeutung, da die zur Verfügung stehende Zeit für die Arbeit mit dem Benutzer oft sehr knapp ist [Mil00]. Technische und organisatorische Voraussetzungen einer Methode sind dann relevant, wenn die Umgebungsbedingungen vor Ort die Durchführung einer Methode verhindern können. Beispielsweise ist es aufgrund interner Firmenrichtlinien in einigen Fabrikhallen nicht erlaubt Videoaufzeichnungen durchzuführen, was die Verwendung darauf basierender Methoden ausschließt. Die ermittelten Eigenschaften werden im Anschluss an die Vorstellung der Methoden diskutiert, um die Auswahl von Methoden für ein konkretes Projekt anzuleiten.

Interviews

Interviews sind eine klassische Methode der Sozial- und Arbeitswissenschaften, um Informationen über eine Arbeitssituation zu erheben. Interviews können auf unterschiedliche Art und Weise durchgeführt werden. Die Techniken unterscheiden

Eigenschaft	**Bewertung**
Zeit Nachbearbeitung	hoher Aufwand
Aufgabenstruktur	sehr gut geeignet
Umgebungseinflüsse	gut geeignet (außer physische Umgebung)
Interaktionsressourcen	schlecht geeignet
Beispiele zeitl.	nicht geeignet
sonst. Voraussetzungen	keine

Tabelle 3.1: Eigenschaften von Interviews

sich vor allem durch die Art der Gesprächsführung. Die Technik, die in der Regel für die Analyse der Arbeitsumgebung eines Benutzers eingesetzt wird, verwendet hauptsächlich offene Fragen und einen wenig strukturierten Interview-Leitfaden. Diese Art der Interviews wird deshalb auch als *wenig strukturiert* bezeichnet [Woo97].

Im Gegensatz zu stark strukturierten Interviews werden keine detaillierten Interviewleitfäden eingesetzt, sondern nur wenig strukturierte Richtlinien. Diese Richtlinien bestimmen vor allem die Vorgehensweise, die bei der Befragung eingesetzt wird, nicht jedoch die spezifischen Fragen. Das Hauptziel eines solchen Interviews ist es einen Überblick über die bestehende Arbeitssituation des Benutzers zu gewinnen. Daher ergeben sich konkrete Fragestellungen erst während des Interviews. Die Benutzer fungieren hier als Domänenexperten und der Beobachter muss versuchen so viel Domänenwissen wie möglich zu gewinnen, damit er selbst die Domäne besser verstehen kann. Theoretisch werden so viele, bis zu 90-minütige Sitzungen, mit einem Benutzer durchgeführt, bis der Beobachter keine neuen Informationen mehr erhält. In der Praxis endet der Prozess oft früher, wenn die zur Verfügung stehenden Ressourcen (Zeit oder Geld) zur Neige gehen [CAF02].

Interviews zeichnen sich durch einen geringen technischen und organisatorischen Aufwand aus, da keine besondere Vorbereitung notwendig ist. Interviews werden in der Regel aufgezeichnet. Die anschließende Transkription stellt daher einen nicht unerheblichen Aufwand dar. Interviews eignen sich sehr gut, um die Aufgabenstruktur des Benutzers zu dokumentieren. Umgebungseinflüsse, Interaktionsressourcen und detaillierte zeitliche Abläufe lassen sich jedoch nur bedingt erfassen. Der Grund dafür ist, dass Domänenexperten oft nicht in der Lage sind, ihre konkreten Aktivitäten präzise zu beschreiben. Gewöhnung führt häufig dazu, dass wichtige Aspekte der Arbeitssituation nicht mehr durch den Benutzer wahrgenommen werden [Woo97]. Um diese Informationen detailliert zu erheben, werden also andere Methoden benötigt. Tabelle 3.1 fasst die Eigenschaften von Interviews zusammen.

Befragung im Kontext

Bei der Befragung im Kontext (engl. contextual inquiry) wird der Benutzer in seinem tatsächlichen Arbeitsumfeld bei der Durchführung seiner Arbeit begleitet. Dabei greift der Beobachter in das Arbeitsgeschehen ein, um Fragen über die Umgebung oder die aktuelle Aktivität zu stellen [BH98]. Hierzu nimmt der Beobachter häufig die Rolle eines Lehrlings ein, während der Domänenexperte die Rolle des Meisters übernimmt. Da sich der Domänenexperte in seiner natürlichen Umgebung befindet, lassen sich auch solche Aspekte der Arbeitssituation erkennen, die dem Benutzer

Eigenschaft	**Bewertung**
Zeit Nachbearbeitung	mittel (hoch mit Videoaufzeichnung)
Aufgabenstruktur	sehr gut geeignet
Umgebungseinflüsse	sehr gut geeignet
Interaktionsressourcen	sehr gut geeignet
Beispiele zeitl.	schlecht geeignet
sonst. Voraussetzungen	Unterbrechung des Arbeitsprozesses *oder* Videoaufzeichnung möglich

Tabelle 3.2: Eigenschaften von Befragung im Kontext

auf Grund von Gewöhnung nicht mehr bewusst sind. Ist eine Unterbrechung des Benutzers nicht möglich, weil der Arbeitsprozess dadurch gestört würde (z. B. bei einer Endoskopieuntersuchung), dann kann auch auf eine Videoaufzeichnung mit anschließender Befragung zurückgegriffen werden. Diese Videoaufzeichnung kann dann später auch für andere Methoden, wie z. B. die Sequenzanalyse, verwendet werden.

Diese Methode eignet sich hervorragend, um die Aufgabenstruktur des Benutzers zu verstehen und zu dokumentieren. Relevante Umgebungseinflüsse und verwendete Interaktionsressourcen lassen sich ebenfalls gut ermitteln. Zeitliche Abhängigkeiten können wiederum nur auf einem sehr groben Level erhoben werden, da der Ablauf durch den Beobachter gestört und somit beeinflusst wird. Aufgrund dieser Beeinflussung müssen die Ergebnisse dieser Methode jedoch mit Vorsicht betrachtet werden, da es passieren kann, dass die Benutzer ihr Verhalten in Anwesenheit des Beobachters verändern [CAF02]. Deshalb sollten die Ergebnisse dieser Methode zusätzlich durch eine andere Methode verifiziert werden. Tabelle 3.2 fasst die Eigenschaften der Methode Befragung im Kontext zusammen.

Sequenzanalyse

Bei der Sequenzanalyse handelt es sich um eine ganze Familie von Methoden [BG05, AG07], deren Ziel es ist, zeitliche Abhängigkeiten zwischen Aufgaben und Ereignissen zu erfassen. Wie bei der Befragung im Kontext wird der Benutzer in seiner natürlichen Arbeitsumgebung beobachtet. Diesmal wird jedoch nicht in den Prozess eingegriffen, damit dieser so realistisch wie möglich ist. Da während der Beobachtungen nicht auf das Wissen des Benutzers zurückgegriffen werden kann, muss der Beobachter bereits über so viel Domänenwissen verfügen, dass er Ereignisse erkennen und klassifizieren kann. Aus diesem Grund müssen einer Sequenzanalyse zwingend andere Methoden vorausgehen, um eine initiale Ereignisliste erstellen zu können. Während der Beobachtung notiert der Beobachter zu welchen Zeitpunkten Aufgaben oder Ereignisse starten und wann diese wieder enden. Mithilfe dieser Daten können später dann die zeitlichen Abhängigkeiten ermittelt werden. Die Genauigkeit dieser Aufzeichnung und der benötigte Aufwand hängen dabei von der gewählten konkreten Beobachtungsmethode ab. Für die Erfassung von Aufgabenstruktur, Umgebungseinflüssen und Interaktionsressourcen ist diese Methode aufgrund ihres Fokus nicht geeignet. Tabelle 3.3 fasst die Eigenschaften zusammen, die allgemein für die Sequenzanalyse gelten. Anschließend werden drei Vertreter dieser Familie mit ihren spezifischen Vor- und Nachteilen vorgestellt.

Eigenschaft	**Bewertung**
Zeit Nachbearbeitung	*variabel*
Aufgabenstruktur	nicht geeignet
Umgebungseinflüsse	nicht geeignet
Interaktionsressourcen	nicht geeignet
Beispiele zeitl.	sehr gut geeignet
sonst. Voraussetzungen	Aufgabenstruktur bekannt *weitere sind variabel*

Tabelle 3.3: Allgemeine Eigenschaften der Sequenzanalyse

Mit Stift und Papier Werden zeitliche Abhängigkeiten nur mit geringer zeitlicher Granularität benötigt, kann auf Stift und Papier zurückgegriffen werden. Dabei werden auf Karopapier untereinander die zu beobachtenden Ereignisse aufgelistet. Jede Spalte steht dann für eine feste Zeitspanne von z. B. 15 Sekunden. Während der Beobachtung wird mit einer Stoppuhr die Zeit gemessen. Tritt ein Ereignis, in dem einer Spalte zugeordneten Zeitraum auf, wird dort eine Markierung gemacht. Diese Variante ist mit sehr geringem organisatorischem Aufwand durchführbar, jedoch nicht besonders genau. Da sich der Beobachter neben der Arbeitssituation auch noch auf die Stoppuhr und die Notizen konzentrieren muss, gehen leicht Ereignisse verloren oder es werden andere Fehler gemacht. Ebenso stellen gleichzeitig auftretende Ereignisse ein Problem dar, da die Dokumentation der beobachteten Ereignisse einen erheblichen Aufwand bedeutet.

Eigenschaft	**Bewertung**
Zeit Nachbearbeitung	mittel (Übertragung in Computer)
Beispiele zeitl.	Auflösung $\sim$ 15 Sekunden, Fehler möglich
sonst. Voraussetzungen	keine

Tabelle 3.4: Eigenschaften der Sequenzanalyse mit Stift und Papier

Als Videoanalyse Die beste Genauigkeit erreicht man, wenn eine vorher erstellte Videoaufzeichnung am Computer analysiert wird. Eine Videoaufzeichnung erlaubt es, Ereignisse auf Sekundenbruchteile genau zu erfassen. Hierbei stellen auch mehrere gleichzeitig auftretende Ereignisse kein Problem dar. Einmal vorhanden können Videoaufzeichnungen auch für andere Zwecke und weitergehende Analysen verwendet werden. Diesen Detailreichtum und die Flexibilität erkauft sich der Beobachter jedoch mit einem erheblichen Vorbereitungs- und Analyseaufwand, da die benötigte Zeit für eine einfache Videoanalyse bereits etwa das Dreifache der Videolänge beträgt (siehe 5.3). Außerdem kann es sein, dass Videoaufzeichnungen im zu beobachtenden Arbeitsumfeld nicht, oder nur schwer möglich sind. Ist der Benutzer zum Beispiel sehr mobil kann es schwer sein eine Kamera so mitzuführen, dass die interessanten Ereignisse jederzeit im Blickfeld sind. In Krankenhäusern wiederum kann es aufgrund der Anwesenheit von Patienten zum Schutz der Privatsphäre verboten sein, Videoaufzeichnungen durchzuführen.

Eigenschaft	Bewertung
Zeit Nachbearbeitung	hoher Aufwand
Beispiele zeitl.	Auflösung <1 Sekunde
sonst. Voraussetzungen	Videoaufzeichnung möglich

Tabelle 3.5: Eigenschaften der videogestützten Sequenzanalyse

Mit Computerunterstützung Die computergestützte Beobachtung ist ein Kompromiss zwischen diesen beiden Varianten. Hier wird ein TabletPC oder PDA verwendet um die Beobachtung in Echtzeit zu annotieren, wobei der Computer die Zeiterfassung übernimmt. Die so gewonnenen Daten liegen dann im Anschluss an die Analyse sofort vor und müssen nicht, oder nur wenig, nachbearbeitet werden. Da keine manuelle Zeiterfassung mehr notwendig ist, steigt die Genauigkeit gegenüber der Verwendung von Stift und Papier deutlich an. Die Genauigkeit einer Videoanalyse lässt sich dennoch nicht erreichen, da nur begrenzt viele Ereignisse gleichzeitig verfolgt werden können. Hierdurch kann es zu Ungenauigkeiten bezüglich der zeitlichen Abfolge und sogar zu falschen Beobachtungen kommen. Allerdings ist diese Art der Beobachtung auch in schwierigen Umgebungen anwendbar, in denen eine Videoaufzeichnung nicht möglich ist.

Für diese Variante der Beobachtung existieren kommerzielle Produkte wie z. B. den Noldus Pocket Observer [nol07]. Dieser ist jedoch nicht auf Erfassung einer Arbeitssituation optimiert, die sehr kurze und auch parallele Ereignisse enthält. Deshalb wurde im Rahmen dieser Arbeit der *TaskObserver* auf Basis eines TabletPC entwickelt, der genau auf diesen Anwendungszweck spezialisiert ist und in Abschnitt 4.5.1 näher erläutert wird. Mithilfe dieses Werkzeugs wurde eine Studie durchgeführt, die eine computergestützte Beobachtung mit einer Videoanalyse vergleicht und den Kompromiss zwischen Genauigkeit und Aufwand quantifiziert. Diese Studie und ihre Ergebnisse werden später im Abschnitt 5.3 genauer erläutert.

Eigenschaft	Bewertung
Zeit Nachbearbeitung	gering (Fehlerkorrektur)
Beispiele zeitl.	Auflösung $\sim$ 3 Sekunden, Fehler möglich
sonst. Voraussetzungen	keine

Tabelle 3.6: Eigenschaften der computerunterstützten Sequenzanalyse

3.3.3 Auswahl geeigneter Methoden

Tabelle 3.7 fasst die Eigenschaften der vorgestellten Methoden zusammen. Es ist leicht ersichtlich, dass keine Methode alleine in der Lage ist, alle relevanten Daten gut zu erfassen. Daher ist eine Kombination verschiedener Methoden erforderlich. Eine Form der Sequenzanalyse ist notwendig um Traces für den zeitlichen Ablauf zu erhalten. Grundlage für eine Sequenzanalyse ist jedoch die Kenntnis der Aufgabenstruktur. Diese kann sowohl durch Interviews als auch durch Befragungen im Kontext gewonnen werden, wobei Letztere die besseren Ergebnisse liefert [BH98]. Die besten Ergebnisse lassen sich daher erzielen, wenn zunächst eine Befragung im Kontext durchgeführt, anschließend Videoaufzeichnungen stattfinden und diese

	Zeitaufwand Nachbereitung	Aufgabenstruktur	Umgebungseinflüsse	Interaktionsressourcen	zeitl. Abfolge	Voraussetzungen
Interviews	hoch	+	+ (1)	-	- -	
Befragung im Kontext (direkt)	mittel	++	++	++	-	(2)
(Video)	mittel	++	++	++	-	(3)
(nur Beobachtung)	gering	+	o	+	-	
Sequenzanalyse (Stift&Papier)	mittel				o (5)	
(Computer)	gering				+ (5)	
(Video)	hoch				++ (5)	(3)

Tabelle 3.7: Zusammenfassung der verschiedenen Methoden. Die Bewertung (- -/-/o/+/++) ordnet die Methoden relativ zueinander. (1)ohne physische Umgebung (2)Unterbrechung des Arbeitsablaufs möglich (3)Videoaufzeichnung möglich (4)grobe Aufgabenstruktur vorhanden (5)Unterschied nur in der Genauigkeit

Videos dann mittels Sequenzanalyse ausgewertet werden. Diese Vorgehensweise bedeutet jedoch einen erheblichen Zeitaufwand und ist nur möglich, wenn die Arbeit des Benutzers unterbrechbar ist und Videoaufzeichnungen erlaubt sind. Gilt eine dieser Bedingungen nicht, müssen Prozessschritte ausgetauscht werden. Abbildung 3.9 zeigt verschiedene Prozesse unter Berücksichtigung dieser Kriterien.

3.4 Prozessschritt 2: Modellierung

Nachdem die erste Phase der Datenerhebung abgeschlossen ist, müssen die gesammelten Informationen vom Beobachter in ein Arbeitssituationsmodell überführt werden. Dazu sind die vier Arbeitsschritte notwendig, die in Abbildung 3.10 dargestellt sind:

1. Konsolidierung der Traces
2. Erzeugung künstlicher Traces
3. Modellierung der Primäraufgabe
4. Modellierung der Computeraufgabe

3.4.1 Konsolidierung der Traces

Zuerst müssen die Ergebnisse der Sequenzanalyse konsolidiert und bereinigt werden. Die für die Sequenzanalyse verwendete initiale Ereignisliste bildet die Grundlage des

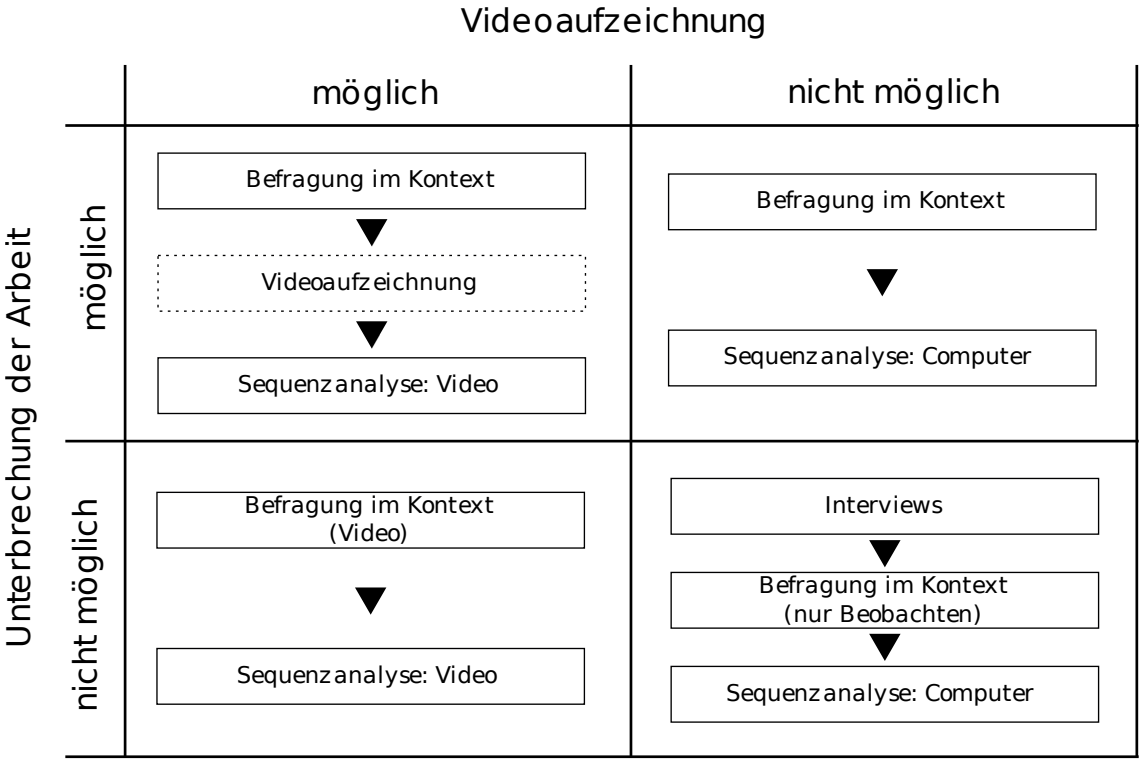

Abbildung 3.9: Empfehlung verschiedener Abläufe in Abhängigkeit der Möglichkeit den Arbeitsprozess zu unterbrechen, und der Möglichkeit Videoaufzeichnungen durchzuführen.

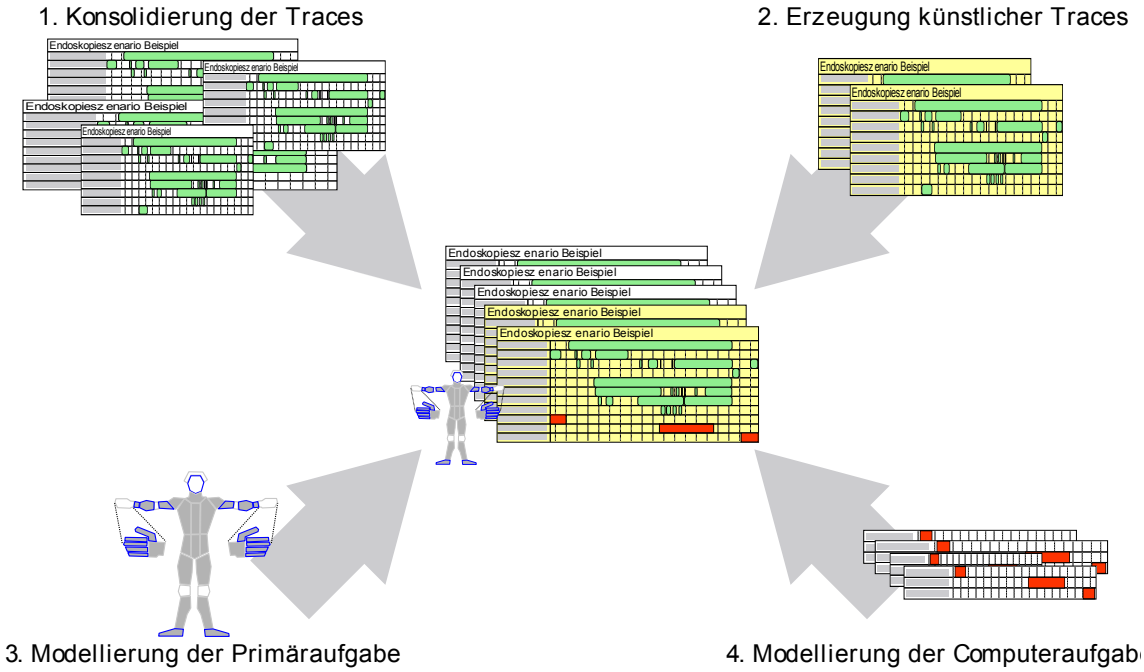

Abbildung 3.10: Prozessschritt 2: Modellierung zur Erstellung eines Arbeitssituationsmodells aus den in Prozessschritt 1 erhobenen Daten.

Primäraufgabenmodells. Da die ursprüngliche Liste während der Beobachtungen durch unvorhergesehene Ereignisse verändert werden kann, müssen die Ereignislisten der einzelnen Beobachtungen konsolidiert werden. Weil die initiale Ereignisliste auf Interviews und Befragungen im Kontext basiert, kann es durchaus vorkommen, dass Änderungen an dieser Liste vorgenommen werden müssen, wenn beispielsweise Abläufe im Vorfeld falsch eingestuft wurden oder Ereignisse nicht erfasst waren. Die einzelnen Traces müssen dann an diese überarbeitete Ereignisliste angepasst werden. Wurden die Datensätze nicht mit Hilfe einer Videoanalyse erstellt, ist es außerdem notwendig Beobachtungsfehler (siehe Kapitel 5.3) zu korrigieren. Dies kann entweder direkt nach der Beobachtung geschehen, wenn die Erinnerungen noch frisch sind, oder später aufgrund von Plausibilitätsüberprüfungen. Die bereinigten Sequenzanalysen werden dann als erste Beispiele in das Modell der Arbeitssituation aufgenommen. Diese Arbeitsschritte werden durch Werkzeuge unterstützt, die in Abschnitt 4.6.1 näher beschrieben werden. Zusätzlich können in diesem Schritt Traces mit redundanten Informationen entfernt werden, um den, für den Entwurf verwendeten Datensatz, so klein wie möglich zu halten.

3.4.2 Erstellung künstlicher Traces

Beobachtungen alleine können in der Regel nicht alle relevanten Situationen erfassen. Aus diesem Grund werden im nächsten Schritt künstliche Traces erstellt. Diese werden mir den Informationen, die während der Interviews und anderen Methoden gesammelt wurden, modelliert. Künstliche Traces bilden Abläufe ab, die selten vorkommen oder unerwünschte Fehlersituationen darstellen.

Beispiel 3

Bei einigen wenigen Endoskopieuntersuchungen kommt es vor, dass ein Argon-Laser eingesetzt wird, um ambulant einen kleinen Eingriff am Magen des Patienten durchzuführen.

Sowohl die Primär- als auch die Computeraufgabe des Arztes kann durch den Einsatz des Lasers beeinflusst werden. Konnte diese Art der Behandlung während der Beobachtungen jedoch nicht festgestellt werden, dann muss sie als künstlich erstellter Trace in das Arbeitssituationsmodell eingefügt werden. Ein künstlicher Trace kann sowohl durch Modifikation aus einem real beobachteten Trace gebildet, als auch vollständig konstruiert werden, falls dies erforderlich ist. Weil künstliche Traces in der Regel zusätzliche Ereignisse enthalten, müssen das Primär- und das Computeraufgabenmodell ebenfalls modifiziert werden. Die Modellierung eines künstlichen Traces ist allerdings nur dann notwendig, wenn der Trace andere Anforderungen an die Interaktionsressourcen des Benutzers stellt als bereits vorhandene Traces.

Beispiel 4

Ist die Bedienung des Argon-Lasers identisch mit der Bedienung anderer Untersuchungsgeräte, die bereits durch Traces berücksichtigt werden, dann ist die Erstellung eines künstlichen Traces nicht notwendig.

Durch die Erstellung künstlicher Traces, ist theoretisch eine vollständige Abdeckung der möglichen Szenarien einer Arbeitssituation möglich. In der Praxis kann diese Vollständigkeit jedoch nur dann erreicht werden, wenn jeder Handgriff des Arbeitsprozesses lückenlos dokumentiert und vorgeschrieben ist. Eine solche Dokumentation liegt jedoch nur selten vor. Ein pragmatischer Ansatz, der auch bei anderen Verfahren eingesetzt wird [BH98] ist es daher, die Datenerhebung so lange fortzusetzen bis keine neuen Informationen mehr gelernt werden können.

3.4.3 Modellierung des Ressourcenbedarfs

Im nächsten Schritt müssen die Ergebnisse der Beobachtungen der Benutzer dazu verwendet werden, um die Primäraufgaben mit Ressourcenprofilen zu modellieren. Dazu wird die in Abschnitt 3.1.1 beschriebene Beeinflussungssprache herangezogen. Die Modellierung dieser Ressourcenprofile geschieht mithilfe des prototypisch implementierten ProjectEditor, der in Abschnitt 4.6.2 näher beschrieben wird.

3.4.4 Makro-Computeraufgaben definieren und zuordnen

Zuletzt muss noch die Computeraufgabe modelliert werden. Dazu werden durch die Benutzerstudien identifizierten Computeraufgaben zunächst zu Makro-Computeraufgaben gruppiert, die Definition 9 entsprechen. Die beobachteten Aufgaben „Befund diktieren“ und „Ausfüllen des Kurzberichts“ könnten beispielsweise zu einer Makro-Computeraufgabe „Untersuchung dokumentieren“ gruppiert werden. Zusätzlich wird jeder der ursprünglichen Aufgaben eine oder mehrere generische Computeraufgaben aus der Bibliothek zugeordnet, die diese Mikro-Computeraufgabe genauer charakterisiert. Enthält die Bibliothek keine geeignete generische Computeraufgabe, dann muss der Bibliothek eine neue generische Computeraufgabe hinzugefügt werden, die dann für zukünftige Projekte ebenfalls zur Verfügung steht.

Im nächsten Schritt müssen die gebildeten Makro-Computeraufgaben konkreten Zeitbereichen in den Traces zugeordnet werden. Diese Zuordnung beschreibt, wann eine Durchführung der jeweiligen Computeraufgabe in einem konkreten Trace notwendig oder wünschenswert gewesen wäre. Die Flexibilität des Benutzers, den Zeitpunkt der Interaktion in einem gewissen Rahmen selbst planen zu können, soll dabei mit berücksichtigt werden. Deshalb ist diese Zuordnung durchaus unscharf und muss keinen genauen Zeitpunkt spezifizieren. Vielmehr stellt die Zuordnung den Zeitbereich dar, in dem eine Bearbeitung der Aufgabe in diesem Trace logisch sinnvoll gewesen wäre. Die Zuordnung dient dazu herauszufinden, welche Interaktionsgeräte für die Durchführung der Makro-Computeraufgabe in Frage kommen.

Beispiel 5

Die Makro-Computeraufgabe „Patientenakte Einsehen“ findet immer vor der eigentlichen Untersuchung statt. Das bedeutet, dass der Arzt seine Hände frei zur Verfügung hat. Für diese Makro-Computeraufgabe können also ganz andere Geräte eingesetzt werden als für eine Makro-Computeraufgabe, die während der Untersuchung durchgeführt werden soll (siehe Abb. 3.11).

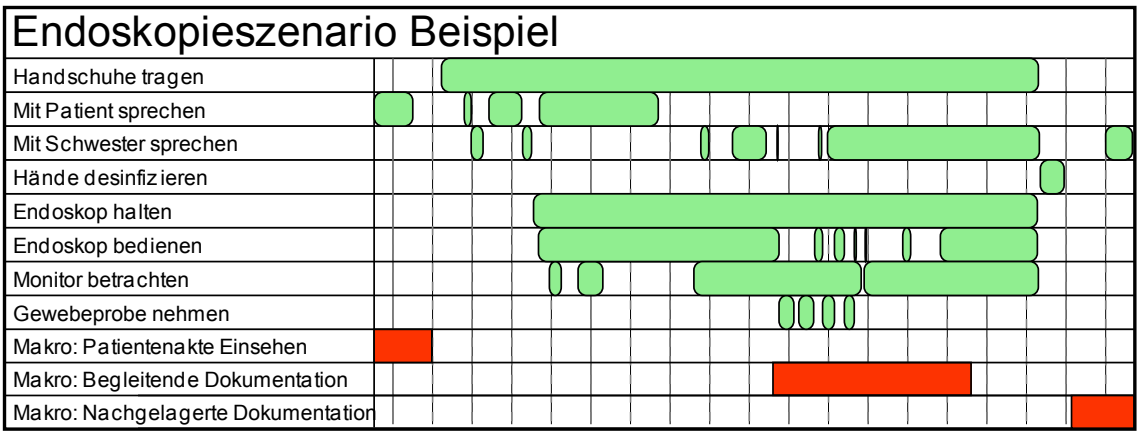

Abbildung 3.11: Zuordnung von Makro-Computeraufgaben zum Ausschnitt eines Traces.

3.5 Prozessschritt 3: Entwurf

Der letzte Prozessschritt der hier beschriebenen Methode unterstützt den *Entwurf* konkreter Wearable-Computing-Systeme. Dabei ermöglicht das Modell der Arbeitssituation dem Designer, die Arbeitssituation besser zu verstehen, die er ja nie mit eigenen Augen erleben konnte. Außerdem erlaubt ihm das Modell bereits in der Entwurfsphase, Anforderungen zu überprüfen und mögliche Systeme hinsichtlich ihrer Kompatibilität mit der Primäraufgabe zu evaluieren. Dieser Prozessschritt gliedert sich wiederum in vier Teilschritte, die in kurzen Iterationszyklen durchlaufen werden, bis ein zufriedenstellender Entwurf gefunden wurde (siehe Abbildung 3.12).

Dieser Teilprozess beginnt mit der *Visualisierung* der Arbeitssituation und einer initialen Analyse bezüglich kompatibler Interaktionsgeräte. Der Designer trifft anschließend, auf Grund dieser Informationen *Entscheidungen* über Änderungen am aktuellen Entwurf. Diesen Entscheidungen entsprechend wird das Modell mit dem WearableDesigner *modifiziert*. Das veränderte Modell wird nun erneut *analysiert* und die nächste Iteration beginnt. Dieser Zyklus wird so lange wiederholt, bis ein zufriedenstellender Entwurf feststeht, der in Form eines Mock-ups oder Prototypen mit Benutzern evaluiert werden muss.

Durch die ständige Analyse der Entwurfsentscheidungen können nachteilige Entwürfe frühzeitig erkannt und korrigiert werden. Deshalb kann auf diese Weise die Anzahl notwendiger Iterationen im Entwurfsprozess reduziert werden, wie die Erfahrungen aus dem Umgang mit der Methode in Abschnitt 5.2 zeigen. Im Folgenden werden die Inhalte der vier Teilschritte näher erläutert. Dieser Teilprozess wird durch zwei Werkzeuge unterstützt. Der *WearableDesigner* visualisiert die zeitlichen Aspekte des Arbeitssituationsmodells sowie die Ergebnisse durchgeführter Analysen. Der prototypisch implementierte *ProjectEditor* ermöglicht den Zugriff auf alle sonstigen Modellelemente. Eine ausführliche Beschreibung beider Werkzeuge befindet sich in Abschnitt 4.6.1.

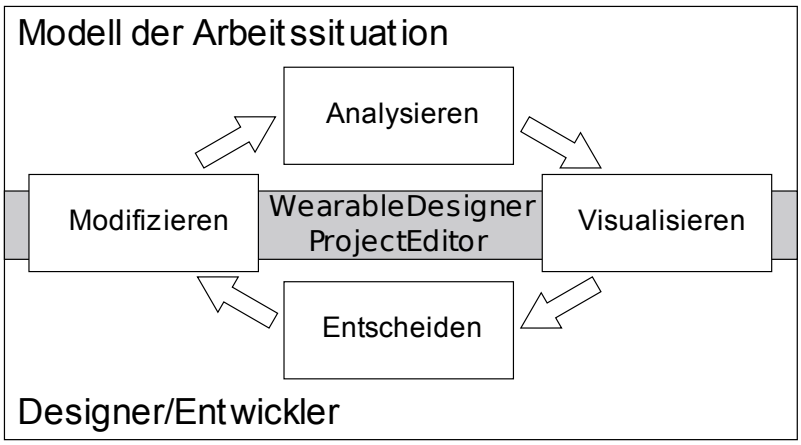

Abbildung 3.12: Prozessschritt 3: Entwurf. Der iterative Entwurfsprozess wird durch den WearableDesigner und den ProjectEditor unterstützt, die dem Designer den Zugriff auf das Arbeitssituationsmodell ermöglichen.

3.5.1 Visualisieren

Das Ziel der Visualisierung ist es, dem Designer die Arbeitssituation so zu präsentieren, dass dieser in der Lage ist, fundierte Entwurfsentscheidungen zu treffen. Damit alle Informationen im Kontext der sie umgebenden Arbeitssituation betrachtet werden können, wurden die Traces als zentrales Visualisierungsobjekt gewählt. Die Visualisierung erfolgt mithilfe der bereits in Beispielen verwendeten Zeitleistendarstellung (z. B. Abb. 3.12). Die Zeit ist bei diesen Diagrammen linear auf der X-Achse aufgetragen. Jede Zeile des Diagramms entspricht einer Art von Ereignis und einzelne Ereignisse werden durch Balken in dieser Zeile dargestellt. Die Länge und Position wird dabei durch Startzeitpunkt und Dauer des Ereignisses bestimmt. Diese Darstellung hat den Vorteil, dass wichtige Merkmale der Arbeitssituation direkt in der grafischen Darstellung kodiert sind und intuitiv erfasst werden können. Diese Merkmale sind die Dauer und relative Position einzelner Ereignisse sowie gleichzeitig stattfindende Ereignisse. Zusätzliche Informationen wie die situationsabhängige Kompatibilität von Interaktionsgeräten sowie die Zuordnung der Makro-Computeraufgaben können, wie in den Abbildungen 3.11 und 3.13 zu sehen, ebenfalls in diese Diagramme integriert werden. Eine detaillierte Beschreibung der möglichen Visualisierungen im WearableDesigner befindet sich in Abschnitt 4.6.1.

3.5.2 Entscheiden

Auf Basis der Visualisierung folgt nun der eigentliche kreative Entwurfsprozess. Hier muss der Designer eine Reihe von Fragen beantworten:

- Wie verändert sich die Primäraufgabe durch das Computersystem?
- Sind Änderungen an der Computeraufgabe sinnvoll?
- Welche Interaktionsgeräte sollen verwendet werden?

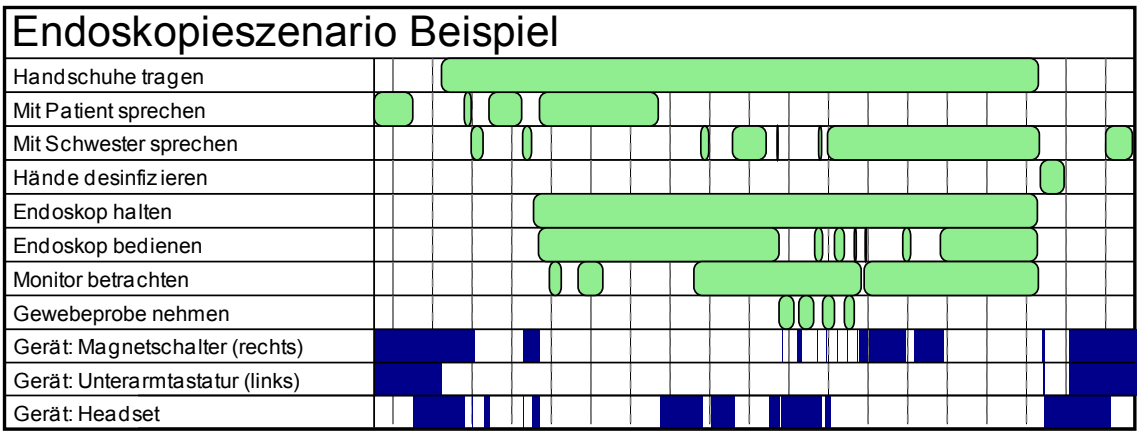

Abbildung 3.13: Darstellung der Gerätekompatibilität integriert in den Trace einer Endoskopieuntersuchung. Blaue Balken markieren Zeitbereiche in denen das jeweilige Interaktionsgerät bedienbar wäre.

Durch die Einführung eines Computersystems kann sich auch die Primäraufgabe verändern. So kann es sein, dass Aktivitäten entfallen, sich ändern oder neu hinzukommen. Diese Entscheidung erfordert jedoch Domänenwissen, das weit über das verwendete Arbeitssituationsmodell hinausgeht. Deshalb müssen diese Entscheidungen vom Designer getroffen werden und können nicht automatisiert werden.

Zudem kann es nötig sein, die Verteilung der Benutzeraufgaben zwischen Primäraufgabe und Computeraufgabe zu bearbeiten. So können zu Beginn des Entwurfsprozesses noch nicht für eine Computerimplementierung vorgesehene Primäraufgaben in Makro-Computeraufgaben umgewandelt werden. Ebenso kann es z. B. aus Kostengründen sinnvoll sein, zunächst für das Computersystem vorgesehene Aufgaben doch nicht mit dem Computersystem zu unterstützen.

Die dritte Entscheidung, die der Designer treffen muss, ist die Auswahl geeigneter Interaktionsgeräte, die als Gerätekonfiguration das zukünftige Computersystem beschreiben. Diese Gerätekonfiguration muss in der Lage sein, alle Makro-Computeraufgaben vollständig zu implementieren, um die Arbeitssituation vollständig unterstützen zu können. Dabei muss nicht jedes Gerät für jede Makro-Computeraufgabe verwendet werden, da nicht alle Interaktionsgeräte immer einsetzbar und auch nicht für alle generischen Computeraufgaben geeignet sind. Es muss aber für jede Makro-Computeraufgabe eine Menge an Interaktionsgeräten geben, die mithilfe von Interaktionsstrategien alle Mikro-Computeraufgaben umsetzen können.

3.5.3 Modifizieren

Anschließend müssen die getroffenen Entwurfsentscheidungen umgesetzt werden. Erstens kann das Arbeitssituationsmodell modifiziert werden, um eine angepasste Aufteilung der Arbeitssituation in Primär- und Computeraufgabe zu reflektieren. Zweitens kann ein Entwurf als Kombination von Interaktionsgeräten modelliert werden, um dessen Eignung für die Arbeitssituation zu evaluieren. Beide Aufgaben werden durch den WearableDesigner und den ProjectEditor unterstützt und werden in Kapitel 4 näher erläutert.

3.5.4 Analysieren

Die Analysephase schließlich ist ein automatischer Prozess, der innerhalb der Modelle stattfindet. Die Ergebnisse der Analysen werden im Arbeitssituationsmodell gespeichert und stehen dem Designer in der Visualisierungsphase der nächsten Iteration zur Verfügung. Es werden zwei automatische Analysen durchgeführt. Erstens wird die Kompatibilität von Interaktionsgeräten ermittelt und zweitens werden mögliche Interaktionsstrategien zur Umsetzung der Computeraufgaben ermittelt. Beide Verfahren werden in Kapitel 4 detailliert beschrieben.

3.6 Zusammenfassung

Dieses Kapitel hat die wesentlichen Konzepte und Modelle dieser Arbeit, zur besseren Unterstützung des benutzerorientierten Entwurfs von Wearable-Computing-Systemen dargestellt. Dabei wurde zunächst ein Arbeitssituationsmodell vorgestellt, das in der Lage ist Wearable Computing spezifische Anforderungen zu erfassen. Dieses Modell vereinfacht zum Einen die Kommunikation mit den Teammitgliedern des Entwurfsteams, und ermöglicht zum Anderen die automatische Analyse der Arbeitssituation. Um diese Aufgaben zu unterstützen, wurde außerdem ein Benutzermodell vorgestellt, das die Simulation der verfügbaren Interaktionsressourcen eines Benutzers unter Berücksichtigung seiner Primäraufgabe ermöglicht. Dazu wurde ein Computersystemmodell entwickelt, das Interaktionsgeräte und Interaktionsstrategien beschreibt, die technischen Komponenten, die für den Entwurf einer Wearable-Computing-Lösung zusammengestellt werden müssen. Existierendes Designwissen kann mit diesen Modellen in Bibliotheken zur Verfügung gestellt werden und ist dem Designer daher auch ohne spezifisches Fachwissen eine Hilfe..

Ferner wurde aufbauend auf diesen Modellen ein Prozess beschrieben, der bestehende benutzerorientierte Entwurfsmethoden um Wearable-Computing-spezifische Lösungen erweitert. Dabei werden drei wesentliche Punkte berücksichtigt. Erstens die Unterstützung des Beobachters bei der Feldforschung in Arbeitssituationen fernab eines Büros. Zweitens die Verbesserung der Kommunikation zwischen Beobachter und Designer, indem die für Wearable Computing relevanten Aspekte der Arbeitssituation zunächst modelliert und dann visualisiert werden. Drittens wird der Designer beim Entwurf konkreter Wearable-Computing-Lösungen unterstützt. Das Modell ermöglicht die automatische Analyse von Anforderungen an die verwendeten Interaktionsgeräte und erlaubt so den Entwurf eines Computersystems, das auf spezifische Situationen innerhalb einer Arbeitssituation angepasst ist.

Im nachfolgenden Kapitel wird nun auf die Details der vorgestellten Modelle eingegangen. Außerdem werden die Algorithmen zur automatischen Analyse erläutert und die Werkzeuge beschrieben, die den Beobachter und den Designer bei der Durchführung des hier beschriebenen Prozesses unterstützen.

Kapitel 4

Methode

Das vorangegangene Kapitel hat die Konzepte, Modelle und Prozesse dieser Arbeit vorgestellt und deren Zusammenhänge erläutert. In diesem Kapitel soll nun auf die Details der Modelle, Algorithmen und Werkzeuge eingegangen werden, die den benutzerorientierten Entwurf von Wearable-Computing-Systemen unterstützen.

Zunächst wird ein Überblick über die entwickelten Modelle gegeben, bevor die konkrete Umsetzung des Benutzermodells, des Arbeitssituationsmodells und des Computersystemmodells beschrieben wird. Darauf folgt die Beschreibung der Algorithmen und Werkzeuge, welche die Grundlage der Entwurfsunterstützung darstellen. Diese werden dabei in den jeweiligen Prozessschritt eingebettet.

4.1 Modellübersicht

Das *Benutzermodell*, das *Arbeitssituationsmodell* und das *Computersystemmodell* wurden zunächst auf der Datenebene modelliert und anschließend um die notwendigen Funktionalitäten erweitert. Zur Modellierung wurde das Eclipse Modeling Framework (EMF) [emf07] verwendet. Das EMF stellt ein Metamodell zur Verfügung, das die Implementierung beliebiger Modelle unterstützt. Aus den Modellen lässt sich Java-Quellcode erzeugen, der dann um Funktionalitäten, wie z. B. die Benutzersimulation, erweitert werden kann. Gleichzeitig wird Quellcode erzeugt, der die Serialisierung von Instanzen dieser Modelle in XML oder in Datenbanken ermöglicht. Da XML eine weit verbreitete Darstellung komplexer Datenstrukturen ist, wurde das Format auch in dieser Arbeit gewählt, um Beispiele konkreter Modellinstanzen darzustellen. Durch die Verwendung des EMF lassen sich die entwickelten Modelle außerdem leicht in andere Eclipse-Anwendungen integrieren und erleichtern so die weitere Verwendung in nachfolgenden Prozessen.

4.2 Benutzermodell

Das Benutzermodell basiert auf einer Analyse verschiedener Interaktionsgeräte aus Forschung und Industrie sowie der in Abschnitt 2.3 vorgestellten Wearable-Computing-Szenarien. Bei der Auswahl der verwendeten Interaktionsgeräte wurde darauf geachtet, dass ein breites Spektrum verschiedener Interaktionsmethoden abgedeckt wird. Konkret wurden die folgenden Geräte untersucht: Lightglove [HH01],

Twiddler [Sta99], FreeDigiter [MAS04], Gesture Pendant [SAAG00], Finger-Ring [Fuk05], am Körper getragenes Touchpad [TGM+99], Acceleration Sensing Glove [PFHP99], GestureWrist [Rek01], Textile Bedienelemente [TMTP02], Magnetschalter [NOKK04], Vibrationsdisplay [TDTA03] und eine Unterarmtastatur [TTG98]. Diese Liste deckt ein breites Spektrum an Methoden zur Mensch-Computer-Interaktion ab.

Das Benutzermodell besteht aus drei Komponenten. Den *Interaktionsressourcen*, einer *Beeinflussungssprache* und der *Benutzersimulation*. Die Interaktionsressourcen beschreiben, analog zu Bürgy's Benutzermodell den Zustand eines Benutzers, sind jedoch ausdrucksstärker. Die Beeinflussungssprache erlaubt die automatisierte Veränderung des Modells und ermöglicht damit die Simulation einer Arbeitssituation. Die Interaktionsressourcen und die Beeinflussungssprache werden nun im Detail beschrieben, während die Benutzersimulation später im Abschnitt 4.7.1 erläutert wird.

4.2.1 Interaktionsressourcen

Nach Definition 5 ist eine Interaktionsressource eine Ressource des Benutzers, die ihm direkt oder indirekt die Interaktion mit seiner Umgebung ermöglicht. Die Hände und die Ohren sind Beispiele für solche Ressourcen. Die Hände ermöglichen dem Anwender Knöpfe zu drücken oder Tastaturen zu bedienen und seine Ohren ermöglichen ihm die Wahrnehmung von akustischen Informationen. Zusätzlich werden hier auch solche Ressourcen als Interaktionsressourcen bezeichnen, die zwar nicht direkt der Interaktion dienen, aber dazu beitragen, dass eine Interaktion auf anderem Wege möglich wird.

Auf der Basis der analysierten Interaktionsgeräte wurden drei relevante Klassen von Interaktionsressourcen identifiziert: *Aktoren*, *Sensoren* und *Körperbereiche*. Aktoren sind alle Möglichkeiten des Benutzers seine Umgebung zu beeinflussen während Sensoren alle Möglichkeiten sind seine Umgebung wahrzunehmen. Körperbereiche sind Regionen am Körper, die entweder dazu verwendet werden können, um Interaktionsgeräte zu befestigen oder die mit einem bestimmten Aktor oder Sensor verknüpft sind. Abbildung 4.1 zeigt einen grafischen Überblick über die gewählte Aufteilung des Körpers in Körperbereiche sowie die Zuordnung von Sensoren und Aktoren, die im Folgenden näher erläutert wird.

Körperbereiche

Im Gegensatz zu herkömmlichen Interaktionsgeräten wie Tastatur und Maus müssen tragbare, im Sinne von anziehbare, Interaktionsgeräte in der Regel am Körper befestigt werden, da der Benutzer sehr mobil ist. Der menschliche Körper bietet allerdings nur begrenzt viel Platz um Gegenstände zu befestigen, vor allem, da Kleidung ebenfalls einen Einfluss auf den verfügbaren Platz hat. Zusätzlich können nicht alle diese Orte verwendet werden, da Interaktionsgeräte an der falschen Stelle den Benutzer stören könnten.

Gemperle u. A. [GKS+98] haben deshalb Kriterien für die Tragbarkeit (engl. wearability) erarbeitet, die zu einer guten Akzeptanz durch den Benutzer führen. Mithilfe dieser Kriterien und umfangreichen Benutzerstudien hat Gemperle Orte am

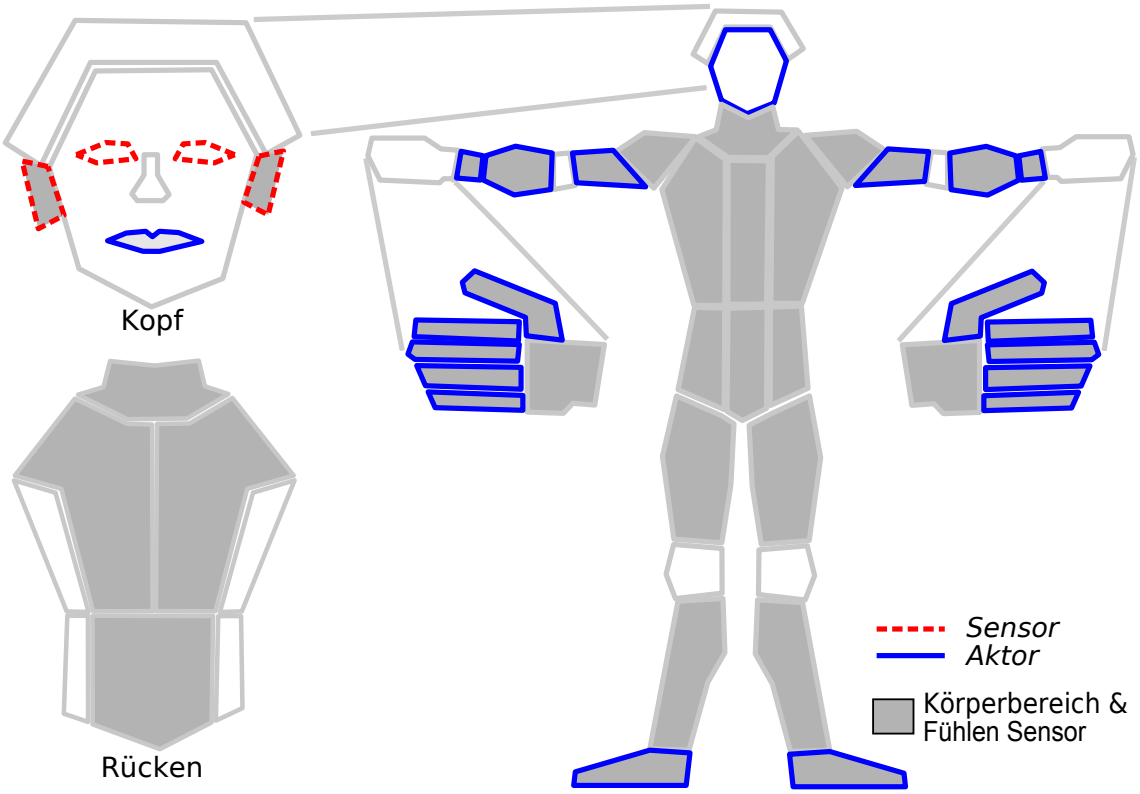

Abbildung 4.1: Grafische Repräsentation des Benutzermodells. Graue Flächen sind im Modell enthaltene Körperbereiche, die auch mit Sensoren oder Aktoren verknüpft sein können.

Körper identifiziert, an denen Geräte befestigt werden können, ohne die natürliche menschliche Bewegung zu stören und an welchen sie als Teil des Körpers wahrgenommen werden, anstatt als separate Entität. Diese Orte dienten als Grundlage des hier beschriebenen Benutzermodells. Gemperle's Modell musste jedoch um einige Orte, wie z. B. den Hals und die Ohren erweitert werden, da er sich mit der Platzierung von etwa PDA-großen Geräten beschäftigt hat, während einige der in Abschnitt 2.2 beschriebenen Geräte deutlich kleiner sind. Durch die kleineren Ausmaße können auch an anderen Befestigungsorten die Kriterien für die Tragbarkeit erfüllt werden. Ein Standard Bluetooth Headset kann z. B. direkt am Ohr befestigt werden, ohne den Benutzer zu behindern. Andere Geräte, wie z. B. das GesturePendant, welches an einer Schnur um den Hals getragen wird, erfüllen die von Gemperle geforderten Kriterien zwar nicht, aber Benutzerstudien haben gezeigt, dass dieses Interaktionsgerät trotzdem benutzbar ist. Abbildung 4.1 zeigt die Aufteilung des Körpers in die Bereiche, die durch die Arbeit von Gemperle und die notwendigen Erweiterungen entstanden sind. Dabei ist jedem dieser Körperbereiche ein Bezeichner zugeordnet (siehe Tabelle 4.1). Diese sind hierarchisch organisiert, sodass die Notation verkürzt werden kann. Beispielsweise bezeichnet *body.left.arm.hand* alle Finger und die Handfläche der linken Hand. Auf diese Weise können mehrere Körperbereiche gleichzeitig angesprochen werden.

Mithilfe dieser Körperbereiche kann nun das Anziehen, Tragen und Ausziehen

Körperbereich	**Bewegung**
body.head.top	ja
body.head.eyes.[left/right]	nein
body.head.mouth	nein
body.head ears.[left/right]	nein
body.neck.[front/back]	nein
body.chest.[front/left/right/back]	nein
body.hip.[front/left/right/back]	nein
body.[left/right].leg.[thigh/lower leg/foot]	ja
body.[left/right].shoulder	nein
body.[left/right].arm.[upper arm/forearm/wrist]	ja
body.[left/right].arm.hand.[palm/little finger/ ring finger/middle finger/index finger/thumb]	ja

Tabelle 4.1: Bezeichner der modellierten Körperbereiche sind hierarchisch organisiert, um die Notation zu verkürzen.

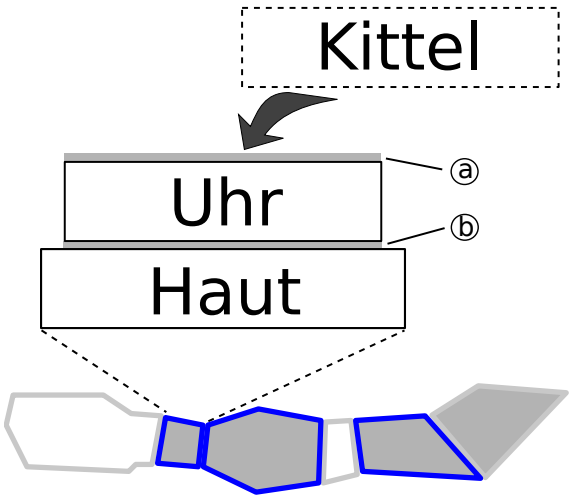

Abbildung 4.2: Objektstapel am Handgelenk. Eine Armbanduhr wird bereits getragen und ein Laborkittel soll angezogen werden. Referenzpunkte für Anforderungen sind die Oberseite eines Elements (a) und die Unterseite (b).

von Kleidungsstücken, Arbeitsgeräten und Interaktionsgeräten modelliert werden. Da Kleidungsstücke in Schichten getragen werden und eine tiefer liegende Schicht in der Regel nicht entfernt werden kann, bevor alle darüberliegenden Schichten ausgezogen wurden, wird dieser Vorgang hier mithilfe eines Objektstapels modelliert. Immer dann, wenn ein neues Kleidungsstück hinzukommt, wird es oben auf den Stapel gelegt (Abb. 4.2). Soll ein Gegenstand entfernt werden, muss dieser oben auf dem Objektstapel liegen. Ist dies nicht der Fall, kommt es in der Simulation zu einem Konflikt. Da die verwendeten Körperbereiche gerade groß genug sind, um ein einzelnes Gerät zu befestigen, stellen sie für diese Anwendung die richtige Granularität dar. Konflikte können außerdem leicht erkannt werden, wenn zwei Geräte in demselben Körperbereich befestigt werden sollen. Jeder Körperbereich ist dabei unabhängig von den Anderen. Die unterste Schicht jedes Körperbereichs ist die Haut, da einige untersuchte Arbeits- und Interaktionsgeräte auf den direkten Kontakt mit der Haut angewiesen sind.

Aktoren

Der menschliche Körper enthält ungefähr 650 Muskeln, die auf unterschiedlichste Weise benutzt werden können, um die Umgebung zu beeinflussen. Allerdings können nur ein paar davon sinnvoll und kontrolliert für die Interaktion mit einem Computer eingesetzt werden. Um die Komplexität des Benutzermodells gering zu halten wurden deshalb auch nur genau diese in das Modell aufgenommen. Die hier untersuchten Interaktionsgeräte verlassen sich meist auf die Benutzung ganzer Muskelgruppen oder Gliedmaßen, um eine Interaktion zu ermöglichen. Die Arme und Hände werden dabei am häufigsten gebraucht, da sie unser vielseitigstes natürliches Werkzeug sind. Aus diesem Grund wurden die Arme und Hände auch in größerem Detail modelliert, als zum Beispiel die Füße. Die Finger und der Daumen jeder Hand wurden zum Beispiel getrennt modelliert, da der Twiddler die Finger zur Texteingabe über Akkorde verwendet und gleichzeitig die Steuerung eines Zeigers mit dem Daumen ermöglicht. Andere Gliedmaßen, die erfolgreich für die Interaktion benutzt wurden, sind die Füße und Bewegungen des Kopfes. Kopfbewegungen können zum Beispiel durch Beschleunigungssensoren erfasst werden, die ein Nicken erkennen und für eine Selektion verwendet werden können. Fußschalter werden unter anderem für Livemusik eingesetzt und stellen ebenfalls eine Interaktionsmöglichkeit dar.

Ein weiterer sehr mächtiger Aktor des Menschen ist die Sprache bzw. die Erzeugung von Tönen. Dieser Aktor ist in Abbildung 4.1 durch den Mund dargestellt. Alle Gliedmaßen, die als Aktor modelliert wurden, sind in dieser Grafik blau umrandet.

Sensoren

Der Mensch hat mindestens fünf Sinne mit denen er externe Reize wahrnehmen kann (Sehen, Hören, Schmecken, Riechen und Fühlen). Von diesen fünf Sinnen können bisher jedoch nur drei sinnvoll für die Interaktion mit dem Computer eingesetzt werden. Die auditive und visuelle Wahrnehmung sind hierbei die gängigsten Wege, um einem Benutzer Informationen zu präsentieren. Die taktile Wahrnehmung hingegen findet bisher nur in wenigen Szenarien Einsatz. Die wohl verbreitetste Form der taktilen Präsentation von Informationen ist der Vibrationsalarm in Mobiltelefonen. Obwohl

zu Forschungszwecken bereits Ausgabegeräte für die gustatorische und olfaktorische Wahrnehmung entwickelt wurden, sind bisher keine sinnvollen Anwendungen in der Mensch-Computer-Interaktion bekannt. Dies ist neben anderen praktischen Gründen vermutlich vor allem auf die Tatsache zurückzuführen, dass diese Ausgabegeräte die präsentierten Informationen nicht schnell genug ändern können.

Neben diesen fünf klassischen Sinnen hat der Mensch noch den Temperatursinn, die Schmerzempfindung, den Gleichgewichtssinn und die Körperempfindung. Diese Sinne können bisher jedoch ebenfalls nicht sinnvoll für die Mensch-Computer-Interaktion eingesetzt werden. Daher enthält das Benutzermodell lediglich den *visuellen*, den *akustischen* und den *taktilen Sinn*.

Der visuelle Sinn wird hier als diskrete Ressource modelliert, die nur von einer Aufgabe gleichzeitig in Beschlag genommen werden kann. Diese Entscheidung wurde aufgrund der Tatsache getroffen, dass Menschen nur eine komplexe Information auf einmal visuell wahrnehmen können. Der Wechsel zwischen verschiedenen Informationsquellen geht zwar schnell, ist aber trotzdem notwendig. Das periphere Sehen kann zwar verwendet werden, um die Aufmerksamkeit des Benutzers auf eine bestimmte Information zu lenken, komplexe Daten wie z. B. Text können auf diesem Wege allerdings nicht vermittelt werden. Da die Präsentation komplexer Daten hier allerdings im Vordergrund steht, kann das Modell entsprechend vereinfacht werden.

Mit dem akustischen Sinn verhält es sich ähnlich wie mit dem visuellen Sinn. Obwohl mehrere Dinge gleichzeitig akustisch wahrgenommen werden können, ist die Aufnahme nur einer komplexen Information möglich. Im Gegensatz zur visuellen Wahrnehmung können 2 komplexe Informationen zwar gleichzeitig wahrgenommen werden, allerdings führt dies in der Regel dazu, dass keine der beiden Informationen interpretiert werden kann. Aus diesem Grund wird die akustische Wahrnehmung analog zur visuellen Wahrnehmung als diskrete Ressource modelliert.

In diesem Punkt unterscheidet sich der taktile vom akustischen und vom visuellen Sinn. Im Gegensatz zu diesen beiden ist der taktile Sinn nämlich nicht an einer Stelle des Körpers gebündelt, sondern in Form der Haut über den ganzen Körper verteilt. Theoretisch können Informationen also am ganzen Körper taktil wahrgenommen werden. Allerdings setzt eine taktile Ausgabe auch ein entsprechendes Gerät voraus, das am jeweiligen Körperteil befestigt werden muss. Aus diesem Grund wird der taktile Sinn, mithilfe der bereits definierten Körperbereiche diskretisiert, da in jedem dieser Bereiche ein Gerät platziert werden kann. Da auf diesem Wege in der Regel nur einfache Informationen präsentiert werden können, ist eine gleichzeitige Aufnahme mehrerer Informationen durchaus denkbar. Wie komplex taktile Ausgaben sein können und wie viele dieser Reize gleichzeitig interpretiert werden können, wird derzeit erforscht [MB06, HB06, HB07]. Allerdings liegen noch keine umfangreichen Studien vor, die dies quantifizieren. Aus diesem Grund sieht das Modell derzeit keine Einschränkung vor, wie viele Körperbereiche gleichzeitig für die taktile Wahnehmung verwendet werden können. Sobald Forschungsergebnisse in dieser Richtung vorliegen, lässt sich das Modell leicht entsprechend erweitern. Bis dahin sollten Designer vorsichtig damit sein, die taktile Wahrnehmung mehrfach parallel anzusprechen.

4.2.2 Beeinflussungssprache

In diesem Abschnitt wird nun eine Beeinflussungssprache entwickelt, die in der Lage ist Änderungen am Zustand eines Benutzers in Bezug auf die Interaktionsressourcen maschinell zu beschreiben.

Dabei werden drei verschiedenen Einflussfaktoren berücksichtigt: das *Tragen von Kleidung und Gegenständen*, *Aktivitäten* und *Anforderungen (engl. requirements)* an den Zustand. Das *Tragen von Gegenständen* modelliert das Anziehen und Ausziehen von Kleidungsstücken und Geräten und damit die Beeinflussung der *Körperbereiche*. *Aktivitäten* modellieren die Benutzung der *Sensoren* und *Aktoren*. *Anforderungen* schließlich beschreiben Beziehungen zwischen den verschieden Modellelementen und schränken deren Verwendung ein. Jeder dieser Einflussfaktoren definiert eine Reihe von *Operatoren*, die das Benutzermodell beeinflussen können. Eine Menge solcher Operatoren bildet dann ein Ressourcenprofil. Insgesamt wurden 10 Operatoren identifiziert, die in den nachfolgenden Abschnitten detailliert beschrieben werden:

- cover
- hear
- see
- speak
- sense touch
- move
- touch
- requireSkinAccess
- mustNotBeCovered
- mustNotTouch

Definition 12 (Operator)

Ein **Operator** *besteht aus einem* Bezeichner *und einer Menge von benannten* Parametern*. Ein Operator wird folgendermaßen beschrieben:*

Bezeichner Allgemeine Beschreibung. **Parameter 1** - Beschreibung Parameter 1. **Parameter 2** - Beschreibung Parameter 2. ...

Tragen von Gegenständen und Kleidung

Für die Modellierung des Tragens eines Gegenstands oder Kleidungsstücks als Operator werden lediglich der Objektname und der betroffene Körperbereich benötigt. Die Objektbezeichnung ermöglicht es anderen Operatoren, das Objekt zu referenzieren und Beziehungen herzustellen.

cover Definiert, dass ein bestimmtes Objekt eine bestimmte Körperregion bedeckt. **where** - Ort an dem das Objekt angebracht wird. **object** - Objektbezeichnung zum späteren Referenzieren.

Beispiel 6

Eine Armbanduhr wird mit der Objektbezeichnung „uhr“ angelegt und befindet sich bei Rechtshändern in der Regel am linken Handgelenk: **cover** (where = „body.left. arm.wrist“, object = „uhr“).

Wird dieser Operator auf eine Simulation angewendet, dann wird die Uhr als oberstes Objekt auf dem Stapel des linken Handgelenks abgelegt. Endet die Verwendung des Profils, wird die Uhr von diesem Stapel wieder entfernt, sofern sie das oberste Element ist. Ein Objekt kann mit mehreren Körperbereichen gleichzeitig assoziiert werden. Ein Datenhandschuh bedeckt beispielsweise alle Finger und das Handgelenk eines Arms.

Aktivitäten

Aktivitäten beschreiben die Verwendung der im Benutzermodell definierten Aktoren und Sensoren. Dementsprechend gibt es einen Operator für jede dieser Ressourcen: *move*, *touch* und *speak* für die Aktoren; *see*, *hear* und *sense touch* für die Sensoren.

Da Sprechen, Sehen und Hören als diskrete Ressource modelliert werden, die nur von einer Aktivität auf einmal verwendet werden können, benötigen diese beiden Operatoren keine weiteren Parameter. Soll das Benutzermodell an dieser Stelle verfeinert werden, können die Operatoren entsprechend erweitert werden. Der Operator für den taktilen Sinn (*sense touch*) benötigt als zusätzlichen Parameter den Körperbereich, in dem ein taktiler Reiz wahrgenommen werden soll.

hear Benutzer benötigt seinen auditiven Sinn vollständig für diese Aktivität.

see Benutzer benötigt seinen visuellen Sinn vollständig für diese Aktivität.

speak Benutzer muss sprechen, um diese Aktivität durchzuführen.

sense touch Benutzer muss in einem bestimmten Körperbereich auf taktile Reize achten. **where** - Körperbereich des taktilen Reizes.

Die Operatoren zur Beschreibung der Bewegung von Körperteilen sind komplexer und benötigen weitere Parameter. Bewegung kann relativ zur aktuellen Position sein, wenn zum Beispiel Beschleunigungssensoren verwendet werden, oder absolut in Bezug auf ein Referenzobjekt wie z. B. einem Knopf, der gedrückt werden soll. In letzterem Fall ist neben der Bewegung auch eine Berührung nötig. Da jedoch nicht jede Bewegung mit einer Berührung einhergeht, werden zu diesem Zweck zwei getrennte Operatoren verwendet. Der Operator *move* zeigt an, dass ein Körperbereich bewegt werden muss, während der Operator *touch* eine Berührung modelliert. Referenzobjekte können andere Körperteile, angezogene Objekte oder Objekte in der Umgebung sein. Die Angabe eines Referenzobjektes ist dann notwendig, wenn das Ziel der Bewegung auch berührt werden muss. Ist das Referenzobjekt dann nicht oben auf dem Stapel der entsprechenden Körperstelle, ist eine Berührung nicht möglich und es kommt zu einem Konflikt.

move zeigt an, dass ein Körperbereich bewegt werden muss. **where** - der betroffene Körperbereich.

touch zeigt an, dass ein Körperbereich einen anderen Körperbereich oder ein Objekt in diesem Körperbereich berührt. **where** - Körperbereich, der die Berührung auslöst. **target** - Körperbereich, der Ziel der Berührung ist *(optional)*. **object** - Objekt, das im Zielbereich berührt werden soll *(optional)*.

Anforderungen

Die dritte Art von Operatoren sind Anforderungen. Sie verändern die Interaktionsressourcen nicht direkt. Stattdessen definieren sie Regeln, die das Benutzermodell erfüllen muss, um in einem gültigen Zustand zu sein. Für die Darstellung der untersuchten Interaktionsgeräte und Szenarien sind drei dieser Anforderungen notwendig. Weitere Anforderungen können dem Modell jedoch leicht hinzugefügt werden.

GestureWrist [Rek01] ist ein Interaktionsgerät, das den direkten Kontakt mit der Haut des Benutzers benötigt. Latexhandschuhe, die während einer Endoskopieuntersuchung getragen werden, müssen ebenfalls direkt auf der Haut sitzen. Die Anforderung, die diese beiden Szenarien beschreibt, nennt sich *requiresSkinAccess*. Der einzige Parameter dieser Anforderung ist des Referenzobjekt, das Kontakt mit der Haut haben muss. Die Anforderung ist dann erfüllt, wenn sich das referenzierte Objekt in jeder Körperregion, in der es angebracht ist, im Stapel direkt über der Haut befindet. Ist dies nicht der Fall, ist die Anforderung nicht erfüllt und es kommt zu einem Konflikt. Die Uhr in Abbildung 4.2 erfüllt dieses Kriterium beispielsweise, der Kittel jedoch nicht.

requiresSkinAccess zeigt an, dass ein Objekt in einem Körperbereich den Kontakt zur Haut benötigt. **where** - Körperbereich, in dem sich das Objekt befindet. **object** - Objekt, das Hautkontakt haben muss.

Andere Interaktionsgeräte dürfen für den Gebrauch nicht durch Objekte verdeckt sein. Das GesturePendant z. B. kann keine Gestenerkennung durchführen, wenn die Kamera, die Handbewegungen filmt, verdeckt ist. Diese Anforderung wird als *mustNotBeCovered* bezeichnet und benötigt ebenfalls ein Referenzobjekt als zusätzlichen Parameter. Im Gegensatz zur Anforderung *requiresSkinAccess* muss das Referenzobjekt sich jedoch oben auf dem Stapel befinden. Würde diese Anforderung für die Uhr in Abb. 4.2 gelten, dann wäre die Anforderung nach Anziehen des Kitteln nicht mehr erfüllt.

mustNotBeCovered zeigt an, dass ein Objekt in einem Körperbereich nicht durch ein anderes Objekt bedeckt sein darf. **where** - Körperbereich, in dem sich das Objekt befindet. **object** - Objekt, das nicht bedeckt sein darf.

Die dritte Anforderung, *mustNotTouch*, ergibt sich aus dem betrachteten Endoskopieszenario. Die Hygienebestimmungen während einer solchen Untersuchung verlangen, dass der Arzt keine Gegenstände anfassen darf, die nicht desinfiziert werden. Diese Regel betrifft nicht-medizinische Geräte, die Kleidung des Arztes sowie Gegenstände der Umgebung. Ausgenommen sind jedoch die für die Untersuchung erforderlichen Utensilien, wie das Endoskop oder Spritzen. Die Anforderung hat daher zwei Parameter. Der erste Parameter beschreibt welcher Körperteil keine Objekte berühren darf, während der zweite Parameter eine Liste von Ausnahmen enthält. Die Bedingungen während einer Untersuchung können also wie folgt modelliert werden:

mustNotTouch fordert, dass ein Körperbereich keinen anderen Körperbereich oder Gegenstand berühren darf, wobei Ausnahmen definiert werden können. **where** - Körperbereich, der nichts berühren darf. **exceptions** - Liste von Objekten und Körperregionen, für die dieser Operator nicht gilt.

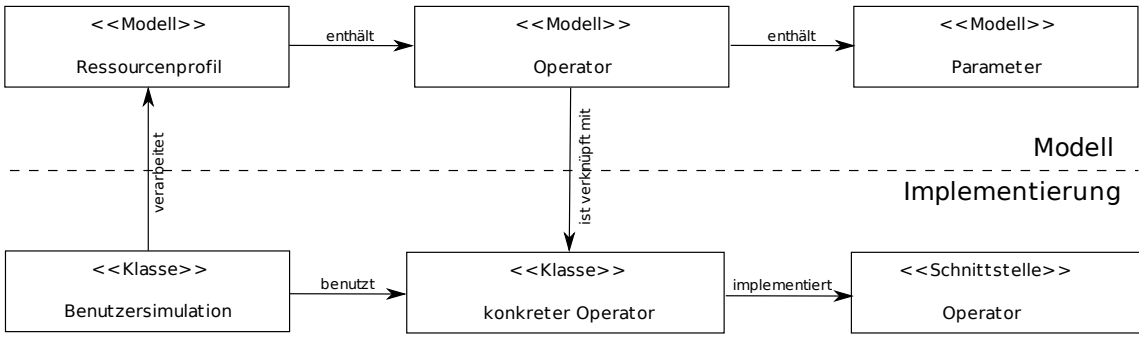

Abbildung 4.3: Kopplung zwischen Modell und Implementierung beim Benutzermodell.

Beispiel 7

Für jede Körperfläche an den Händen wird die Anforderung mustNotTouch *angelegt. Die Liste der Ausnahmen enthält das Endoskop, Spritzen und auch die Handschuhe (gegenseitige Berührung der Hände):* **mustNotTouch** (where = „body.left.arm.hand“, exceptions = „endoskop;spritze;handschuh“) **mustNotTouch** (where = „body.right.arm.hand“, exceptions = „endoskop;spritze;handschuh“)

Während der Simulation wird für jede dieser Anforderungen zunächst überprüft, ob der entsprechende Körperbereich mit einem Objekt in Berührung steht (*touch*). Ist dies der Fall, wird das Objekt, bei Körperflächen das oberste des Stapels, mit der Liste der Ausnahmen verglichen. Ist es hier nicht enthalten, ist die Anforderung nicht erfüllt.

4.2.3 Ressourcenprofil

Gemäß Definition 6 beschreibt ein *Ressourcenprofil* den Bedarf an Interaktionsressourcen einer Aktivität, eines Umgebungseinflusses oder eines Interaktionsgerätes durch Anforderungen und benötigte Interaktionsressourcen. Die Anforderungen und benötigten Interaktionsressourcen können mithilfe der eben beschriebenen *Operatoren* modelliert werden. Ein Ressourcenprofil ist also eine Menge an Operatoren. Dabei kann die Reihenfolge der Operatoren eine Rolle spielen, wenn z. B. zwei Kleidungsstücke in einer bestimmten Reihenfolge angezogen werden müssen. Konkrete Beispiele für Ressourcenprofile folgen in den Abschnitten 4.3.1 und 4.4.1.

Das Ressourcenprofil ist ein zentraler Bestandteil der hier beschriebenen Modellierung und wird sowohl im Primäraufgabenmodell als auch im Interaktionsgerätemodell verwendet, um deren Einfluss auf den Benutzer zu beschreiben.

4.2.4 Implementierung

Die Konzepte des Benutzermodells sind wie in Abbildung 4.3 gezeigt implementiert. Ressourcenprofile, Operatoren und Parametern sind Modellelemente, die verschiedene Anforderungen beschreiben. Jeder Operator ist dabei mit einer spezifischen Implementierung verknüpft, die konkrete Auswirkungen auf den Zustand des Be-

nutzers umsetzt. Diese implementieren jeweils die Operator-Schnittstelle, die von der Benutzersimulation verwendet wird, um die einzelnen Operatoren eines Ressourcenprofils auszuführen.

Sollen neue Interaktionsressourcen, wie z. B. die olfaktorische Wahrnehmung hinzugefügt werden muss, zum Einen die Benutzersimulation modifiziert werden und zum Anderen muss ein spezifischer Operator implementiert werden, der den Zustand dieser Interaktionsressource modifizieren kann. Zusätzliche Anforderungen erfordern keine Veränderung an der Benutzersimulation. Hier muss lediglich ein neuer Operator angelegt werden, der Beziehungen zwischen verschiedenen Interaktionsressourcen überprüft.

4.2.5 Zusammenfassung

Das Benutzermodell bildet den Benutzer durch *Interaktionsressourcen* ab, die sowohl für die Modellierung einer Primäraufgabe, als auch für die Modellierung von Interaktionsgeräten wichtig sind. Das Benutzermodell besteht dabei aus zwei Teilen. Einer statischen Zustandsbeschreibung eines Benutzers, die die zu einem bestimmten Zeitpunkt verfügbaren Interaktionsressourcen abbildet und einer *Beeinflussungssprache*, die mithilfe verschiedener *Operatoren* Einfluss auf diesen Zustand nehmen kann. Die Operatoren bilden die Wahrnehmung des Benutzers ab, seine Aktivitäten sowie externe Anforderungen. Operatoren werden in *Ressourcenprofilen* zusammengefasst, die jeweils den Einfluss eines zusammenhängenden Ereignisses auf den Benutzer beschreiben. Diese Ressourcenprofile werden im Folgenden verwendet um Interaktionsgeräte und Primäraufgaben zu modellieren.

4.3 Arbeitssituationsmodell

Eine Arbeitssituation besteht wie in Abschnitt 3.1.2 beschrieben aus den drei Teilmodellen: *Primäraufgabenmodell*, *Computeraufgabenmodell* und *Modell des zeitlichen Ablaufs*. Im Folgenden werden diese Teilmodelle nun detailliert beschrieben.

4.3.1 Primäraufgabenmodell

Das Primäraufgabenmodell beschreibt die verschiedenen Elemente der Primäraufgabe: Aktivitäten des Benutzers und Umgebungseinflüsse. Beide Arten werden innerhalb des Modells gleichwertig behandelt und im Folgenden als Primäraufgaben bezeichnet.

Um den Ressourcenbedarf einer Primäraufgabe zu beschreiben, wird ihr ein Ressourcenprofil zugeordnet, das die Aktivitäten und Anforderungen, die sich durch die Aufgabe ergeben modelliert. Der Ressourcenbedarf wird dabei über die Dauer der Aufgabe als konstant angenommen (siehe Regel 1). Varianten im Ressourcenbedarf müssen deshalb in separaten Aufgaben modelliert werden. Zusätzlich kann jede Primäraufgabe einer Kategorie zugeordnet werden. Kategorien sind ein Hilfsmittel für den Beobachter oder Designer, um die Primäraufgaben nach verschiedenen Kriterien zu ordnen. Beispielsweise können verschiedene Rollen unterschieden oder Umgebungseinflüsse besonders hervorgehoben werden.

Beispiel 8

Das Ressourcenprofil der Aufgabe, einen Patienten mit den Händen untersuchen, *enthält vier Operatoren. Beide Arme müssen bewegt (***move***-Operator) werden, um den Patienten abtasten zu können. Deshalb berühren die Hände zusätzlich den Patienten (***touch***-Operator):*

```
<primarytask name="Patient untersuchen"
    category="//@categories.0" id="3">
<resourceProfile>
  <operator name="move">
    <properties key="where" value="body.left.arm"/>
  </operator>
  <operator name="move">
    <properties key="where" value="body.right.arm"/>
  </operator>
  <operator name="touch">
    <properties key="where" value="body.left.arm.hand"/>
    <properties key="object" value="patient"/>
  </operator>
  <operator name="touch">
    <properties key="where" value="body.right.arm.hand"/>
    <properties key="object" value="patient"/>
  </operator>
</resourceProfile>
</primarytask>
```

4.3.2 Computeraufgabenmodell

Makro-Computeraufgaben beschreiben die Aufgaben des Benutzers, die in Zukunft mithilfe des Computersystems durchgeführt werden sollen. Jede Makro-Computeraufgabe erhält einen domänenspezifischen Namen und eine Beschreibung, die anderen Teammitgliedern die Bedeutung der Makro-Computeraufgabe erläutern soll. Außerdem enthält eine Makro-Computeraufgabe gemäß Definition 9 eine oder mehrere Mikro-Computeraufgaben.

Eine *Mikro-Computeraufgabe* ist eine Instanz einer generischen Computeraufgabe. Ihr wird jedoch zusätzlich eine natürlichsprachliche und domänenspezifische Beschreibung hinzugefügt. Diese Beschreibung ermöglicht es dem Beobachter oder Designer, den Zweck einer Mikro-Computeraufgabe zu dokumentieren. Die generische Computeraufgabe wird aus einer Bibliothek ausgewählt, um Designwissen aus vorherigen Wearable Computing Systemen wiederverwenden zu können. Generische Computeraufgaben werden in Abschnitt 4.4.2 näher erläutert.

Beispiel 9

Eine der Makro-Computeraufgaben im Endoskopieszenario ist bspw. das „Einsehen der Patientenakte“. *Diese Computeraufgabe ermöglicht es dem Arzt, während der Untersuchung auf Dokumente vergangener Untersuchungen zuzugreifen. So kann er etwa Fotos von krankem Gewebe mit vorherigen Fotos vergleichen. Um diese Makro-Computeraufgabe zu implementieren, wird sie in drei Mikro-Computeraufgaben unterteilt:* „Dokument auswählen“, „Dokument ansehen“ *und* „Schließen des aktuellen Dokuments“. *Jede dieser Mikro-Computeraufgaben ist eine Instanz einer generischen Computeraufgabe, die mit domänenspezifischen Informationen versehen wurde.* „Dokument auswählen“ *ist beispielsweise eine Instanz der generischen Computeraufgabe* „Auswahl aus einer Liste“. *Die generischen Computeraufgaben der anderen beiden Mikro-Computeraufgaben sind* „Navigation in einem Dokument“ *und* „Aktion“. *Die Modellierung würde in XML dann wie folgt aussehen:*

```
<macrocomputertask name="Einsehen der Patientenakte"
   description="Diese Computeraufgabe ermöglicht es ...">
  <microcomputertask name="Dokument auswählen"
     generic="Auswahl aus einer Liste"
     description="Der Arzt wählt ein Dokument..."/>
  <microcomputertask name="Dokument ansehen"
     generic="Navigation in einem Dokument"
     description="..."/>
  <microcomputertask name="Dokument schließen"
     generic="Aktion"
     description="..."/>
</macrocomputertask>
```

4.3.3 Modell des zeitlichen Ablaufs

Der zeitliche Ablauf einer Arbeitssituation wird, wie in Abschnitt 3.1.2 beschrieben, exemplarisch durch eine Menge von Traces modelliert. Jeder Trace enthält einen Zeitstempel, der den absoluten Beginn des Beispiels bezeichnet sowie eine Beschreibung, die z. B. den Ort der Beobachtung und die beobachtete Situation näher beschreibt.

Des Weiteren enthält jeder Trace *Ereignisse*, die Vorkommnisse bestimmter Primär- oder Computeraufgaben innerhalb des modellierten Zeitraums beschreiben. Ein Ereignis wird dabei durch zwei Zeitstempel und das beschriebene Element definiert. Die Zeitstempel definieren den Beginn und das Ende des Ereignisses relativ zum Beginn des Traces. Zusätzlich können einem Ereignis noch Notizen hinzugefügt werden, um Besonderheiten hervorzuheben. Mehrere Ereignisse derselben Primäraufgabe dürfen sich dabei nicht überschneiden, da eine Primäraufgabe nicht mehrfach parallel auftreten kann. Jeder Trace kann zusätzlich ein Ressourcenprofil enthalten, das den initialen Benutzerzustand dieses Traces definiert. Dies ist vor allem dann notwendig, wenn der Benutzer Arbeitskleidung trägt, die möglicherweise die Benutzung von Interaktionsgeräten beeinträchtigt.

Zusätzlich wird zwischen verschiedenen Typen von Traces unterschieden. *Beobachtete Traces* resultieren aus direkten Beobachtungen und wurden nur bearbeitet,

um Beobachtungsfehler zu korrigieren. *Künstliche Traces* hingegen wurden aufgrund von Interviews oder einer Befragung im Kontext erstell und stellen Situationen dar, die nicht direkt beobachtet werden konnten. Weitere Typen dienen dazu automatisch erstellte aggregierte Traces, die nur der Visualisierung dienen, von originären Traces der Arbeitssituation zu unterscheiden.

Beispiel 10

Dies ist der Trace einer Endoskopieuntersuchung. Zusätzliche Notizen, die bestimmte Ereignisse erläutern, werden direkt in das Modell integriert:

```
<traces name="coloscopy examination thursday morning"
      type="observed" startTime="1158824368545">
  <events element="6" startTime="55000" stopTime="77000"/>
  <events element="26" startTime="119161" stopTime="127103"/>
  <events element="5" startTime="127874" stopTime="135916"/>
  <events element="6" startTime="136286" stopTime="151999"
      description="doctor has problems inserting the endoscope"/>
  <events element="5" startTime="152620" stopTime="156876"/>
  ...
</trace>
```

4.3.4 Implementierung

Abbildung 4.4 zeigt die Implementierung des Arbeitssituationsmodells. Zeitliche Abhängigkeiten werden in diesem Modell mithilfe von Traces implementiert. Ein Trace enthält dabei eine Menge von Ereignissen, die jeweils einem Element der Primäraufgabe zugeordnet sind, das sie repräsentieren. Elemente können dabei entweder Teil der Primäraufgabe sein oder eine Makro-Computeraufgabe repräsentieren. Primäraufgaben enthalten ein Ressourcenprofil, das die Anforderungen der Primäraufgabe an den Benutzer spezifiziert. Makro-Computeraufgaben hingegen enthalten Mikro-Computeraufgaben, welche die genauen Anforderungen der Makro-Computeraufgaben definieren. Die Mikro-Computeraufgaben verweisen auf die generischen Computeraufgaben, die im nachfolgenden Abschnitt beschrieben werden.

4.3.5 Zusammenfassung

Das Arbeitssituationsmodell bildet zunächst mithilfe von *Ressourcenprofilen* die *Primäraufgabe* des Benutzers ab. Dabei können durch die im Benutzermodell definierten *Operatoren* sowohl manuelle Aktivitäten des Benutzers (z. B. Steuerung des Endoskops) als auch rechtliche Rahmenbedingungen (z. B. Arzt darf keine unreinen Gegenstände berühren) berücksichtigen.

Weiterhin wurde ein *Computeraufgabenmodell* definiert, das die Computeraufgabe des Benutzers durch *Mikro-* und *Makro-Computeraufgaben* spezifiziert. Mikro-Computeraufgaben bestehen dabei aus einer *generischen Computeraufgabe*, die eine abstrakte Interaktion des Benutzers mit dem System beschreibt und einer Domänenspezifischen Bezeichnung. Makro-Computeraufgaben hingegen gruppieren Mikro-Computeraufgaben, die zeitlich zusammenhängend durchgeführt werden müssen.

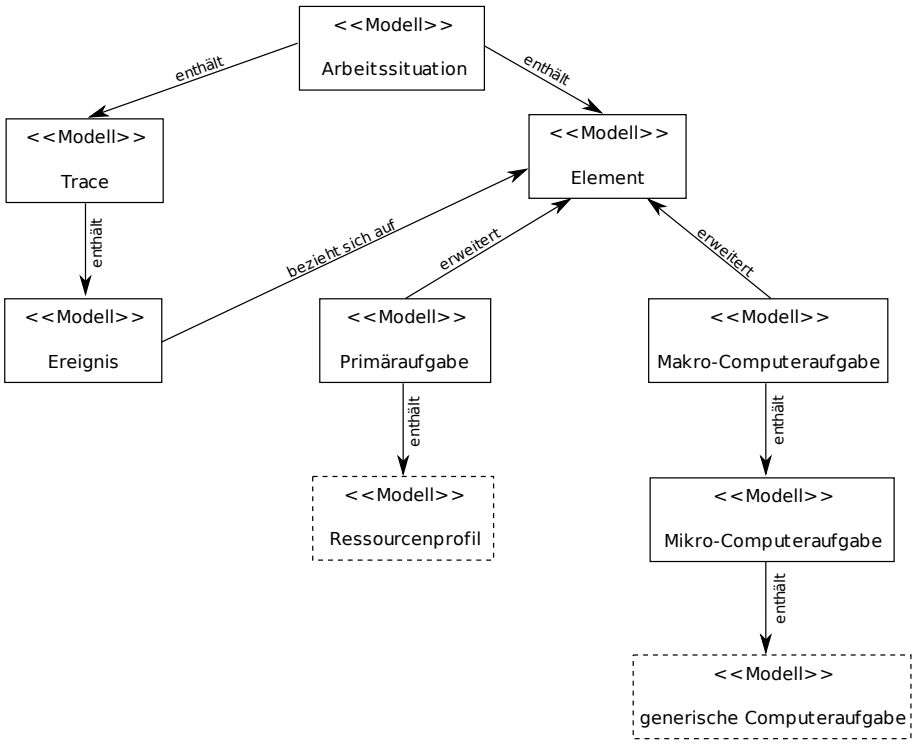

Abbildung 4.4: Beziehungen zwischen Modellelementen des Arbeitssituationsmodells.

Beide Aspekte werden durch ein *Modell des zeitlichen Ablaufs* in zeitlichen Zusammenhang gebracht. Dazu wurden *Traces* definiert, die beobachtete oder künstliche Abläufe in Form von *Ereignissen* modellieren. Jedes Ereignis ist dabei entweder einer Primär- oder Makro-Computeraufgabe zugeordnet. Im Folgenden wird nun näher auf das Computersystemmodell eingegangen, das die Komponenten eines Wearable-Computing-Systems beschreibt.

4.4 Computersystemmodell

Wie in Abschnitt 3.1.3 beschrieben, besteht das Computersystemmodell aus *Interaktionsgeräten* und *Interaktionsstrategien*. Beide Teilmodelle werden nun detailliert beschrieben und mit den Modellen der Arbeitssituation und des Benutzers in Beziehung gesetzt. Das Computersystemmodell enthält außerdem die *Bibliotheken* der Interaktionsgeräte, Interaktionsstrategien und generischen Computeraufgaben, die Designwissen aus vorherigen Projekten bereithalten.

4.4.1 Gerätemodell

Das *Gerätemodell* beschreibt die *Gerätefähigkeiten* eines physischen Interaktionsgerätes, sowie dessen Auswirkungen auf das Benutzermodell, die *Benutzeranforderungen*. Im Folgenden wird zunächst die Umsetzung der Benutzeranforderungen erläutert, bevor die Modellierung der Gerätefähigkeiten im Detail beschrieben wird.

Anschließend wird anhand des Gesture Pendant gezeigt, wie die beiden Aspekte in der Praxis zusammenspielen.

Benutzeranforderungen

Bei den Benutzeranforderungen eines Interaktionsgerätes wird zwischen dem *passiven* und dem *aktiven Ressourcenbedarf* unterschieden. Der passive Ressourcenbedarf wird durch ein Ressourcenprofil modelliert, das direkt mit dem Interaktionsgerät verknüpft ist. Der aktive Ressourcenbedarf kann bei Interaktionsgeräten mit mehreren Funktionen jedoch unterschiedlich ausfallen. Deshalb können zu diesem Zweck mehrere aktive Ressourcenprofile definiert werden. Jedes dieser Ressourcenprofile modelliert dabei die Verwendung einer solchen Funktion. Nutzen verschiedene Funktionen des Interaktionsgerätes dieselben Ressourcen des Benutzers, dann kann hier auch dasselbe Ressourcenprofil referenziert werden.

Gerätefähigkeiten

Das zentrale Konzept der Gerätefähigkeiten ist der *Datenkanal*. Datenkanäle entsprechen den einzelnen Sensoren oder Aktoren eines Interaktionsgerätes. Datenkanäle können daher sowohl der Ein- als auch der Ausgabe dienen. Da Datenkanäle in diesem Modell eindimensional definiert sind (siehe Definition 10), werden *Kanalgruppen* eingeführt, um Beziehungen zwischen mehreren Datenkanälen beschreiben zu können.

Jeder Datenkanal ist mit einer Reihe von Eigenschaften versehen, die sein Verhalten sowie das Datenformat der übermittelten Daten genauer beschreiben. Eigenschaften, die so bisher in keiner der zitierten Arbeiten Verwendung finden sind als *neu* gekennzeichnet. Die Eigenschaften, die für alle Arten von Datenkanälen gültig sind, werden im Folgenden kurz erläutert:

Name - Eindeutiger Name des Datenkanals innerhalb der Gerätebeschreibung. Wird verwendet, um einen bestimmten Kanal zu referenzieren.

Semantik (neu) - In dieser Eigenschaft kann ein Bezeichner festgelegt werden, der die Bedeutung des Datenkanals aus Sicht des Benutzers näher spezifiziert. Es können z. B. Informationen über die Orientierung abgelegt werden, um die Achsen einer Maus als „horizontal“ und „vertikal“ zu unterscheiden. Zusätzlich kann diese Eigenschaft verwendet werden um Bezeichner auf dem Interaktionsgerät festzuhalten. Die Lautstärkeknöpfe einer Fernbedienung können also als „lauter“ und „leiser“ gekennzeichnet werden und zusätzlich abstrakter auch als „hoch“ und „runter“. Muss eine Anwendung die Lautstärke regulieren, so ist dies ein Hinweis, dass diese Knöpfe besser geeignet sind als andere. Statt der verwendeten einfachen Bezeichner könnte an dieser Stelle auch eine Ontologie definiert werden, worauf im Rahmen dieser Arbeit jedoch zunächst verzichtet wurde.

Ereignisse (neu) - Datenkanäle können neben Datenwerten auch Ereignisse senden, um Änderungen ihres Zustands anzuzeigen, die nicht mit einer Datenänderung einhergehen. Einige Geräte sind beispielsweise in der Lage zu erkennen,

wann der Benutzer mit der Interaktion beginnt, bevor eine Wertänderung geschieht. Es gibt z. B. Schieberegler die feststellen können ob der Benutzer den Regler berührt oder nicht. In diesen Fällen kann das Gerät ein *Interaktionsbeginn*-Ereignis senden. Dasselbe gilt für das Ende der Interaktion (*Interaktionsende*). Der Parameter *Ereignisse* enthält eine Liste aller Ereignisse, die von diesem Datenkanal gesendet werden. Interaktionsstrategien können diese Information nutzen, um eine effizientere Interaktion zu ermöglichen.

Ressourcenprofil (neu) - Hier kann auf ein Ressourcenprofil verwiesen werden, das den aktiven Ressourcenbedarf dieses Kanals beschreibt. Benötigen mehrere Kanäle, die gleichzeitig verwendet werden können, dieselben Ressourcen, dann verweisen sie auf dasselbe Profil. Werden zwei Kanäle mit demselben Profil gleichzeitig verwendet wird das Profil nur einmal an die Simulation übergeben.

Privat - Datenkanal kann nur vom Benutzer selbst beeinflusst oder wahrgenommen werden. Ein HMD ist beispielsweise privat, während ein Wandbildschirm öffentlich ist.

Neben diesen allgemeinen Eigenschaften, die für alle Datenkanäle relevant sind, gibt es je nach Typ des Datenkanals weitere Eigenschaften. Typen von Kanälen sind der *Audiokanal*, der *Codewortkanal*, der *Wertekanal* und der *Grafikkanal*. Während der Grafikkanal nur für die Ausgabe verwendet werden kann, werden die anderen Arten für die Beschreibung von Ein- und Ausgabe verwendet. Jeder dieser Kanaltypen definiert zusätzliche Eigenschaften, die zur Beschreibung des Kanals notwendig sind. Diese sind in den nun folgenden Abschnitten erläutert.

Wertekanal

Wertekanäle werden für die Beschreibung klassischer Interaktionskomponenten verwendet, die Sensoren benutzen, um Aktivitäten des Benutzers in einen kontinuierlichen Wert zu übersetzen. Knöpfe, Schieberegler, Computermäuse und Joysticks sind Beispiele für Interaktionsgeräte dieser Art.

Die hier verwendete Beschreibung dieser Wertekanäle basiert auf Forschung aus den 80er und frühen 90er Jahren [CMR91, Bux83, LP93], sowie der USB-HID Spezifikation [Axe05], die viele dieser Konzepte übernommen hat. Basierend auf diesen Vorarbeiten werden für diese Art von Datenkanal eine Reihe von Eigenschaften definiert, die sowohl Ursache als auch Verhalten der generierten Daten beschreiben. Neu sind in diesem Zusammenhang jedoch Informationen über ergonomische Faktoren, wie z. B. die Genauigkeit, die ein Mensch bei der Interaktion des Gerätes typischerweise erreichen kann.

Beispiel 11

Ein Schieberegler wird durch einen Wertekanal beschrieben, der absolute Positionswerte einer Translation liefert. Er ist nicht volatil, hat keinen bevorzugten Zustand und bei Erreichen des Maximums erfolgt auch kein Umspringen auf das Minimum.

relativ - Unterscheidet zwischen Kanälen, die Wertänderungen absolut oder relativ übertragen. Eine Maus überträgt Wertänderungen relativ, während ein Schieberegler absolute Positionsdaten sendet.

Wertebereich(min/max) - Wertebereich der Daten, die auf diesem Kanal gesendet werden. Bei absoluten Kanälen ist dies der tatsächliche Wertebereich, bei relativen Kanälen der gültige Bereich für Änderungen.

physikalische Größe - Gibt die physikalische Größe an, die den Werten des Kanals zugrunde liegt. Hier wird zwischen *Position*, *Geschwindigkeit* und *Beschleunigung* unterschieden. Zusätzlich lässt sich zwischen *Translation*, *Rotation*, *Kraft* und *Drehmoment* unterscheiden.

volatil - Ein Kanal, der volatil ist, behält seinen Wert nicht, wenn der Benutzer die Interaktion beendet. Der erste Wert einer nachfolgenden Interaktion ist also unabhängig vom letzten gesendeten Wert. Ein Touchpad überträgt zum Beispiel keine Werte mehr, sobald der Finger entfernt wird. Die nächste Interaktion kann dann irgendwo auf dem Touchpad erfolgen.

menschliche Auflösung (neu) - Dieser Wert bestimmt die Größe eines Intervalls, das ein Mensch mit dem Interaktionsgerät effektiv selektieren kann. Der technische Wertebereich eines 6 cm langen Schiebereglers ist beispielsweise 1 bis 800. Ein Mensch ist jedoch nicht in der Lage einen dieser Werte zuverlässig zu selektieren. Ist der Parameter *menschliche Auflösung* gleich 100, dann ist ein Mensch in der Lage den Bereich 1 bis 100 zuverlässig anzuwählen. Der Schieberegler könnte demnach also in 8 Bereiche unterteilt werden.

menschliche Änderungsgeschwindigkeit (neu) - Durch diesen Wert wird die Geschwindigkeit bestimmt, mit der ein Mensch den Wertekanal beeinflussen kann. Die Einheit ist der Betrag der maximalen Wertänderung, die ein Mensch in einer Sekunde durchführen kann. Eine Maus kann beispielsweise in einer Sekunde um mehrere Zentimeter bewegt werden, was einer Werteänderung von mehreren Tausend Einheiten entspricht. Das Mausrad kann in derselben Zeit hingegen nur um wenige Einheiten bewegt werden.

bevorzugter Zustand - Die Werte eines Kanals mit einem bevorzugten Zustand kehren nach Ende der Interaktion immer automatisch zu diesem Wert zurück. Ein Joystick kehrt beispielsweise immer in die Mittelstellung zurück, ein Drucksensor auf die Nullposition.

tote Zone - Falls ein Datenkanal einen bevorzugten Zustand hat, dann kann es sein, dass der tatsächlich eingenommene Zustand von diesem abweicht. Die tote Zone definiert dann den Bereich um den bevorzugten Zustand, der ebenfalls als bevorzugter Zustand interpretiert werden kann. Ist keine tote Zone angegeben, dann wird der bevorzugte Zustand exakt eingenommen. Beispiel: Ein Joystick, der per Feder in die Mittelstellung zurückgezogen wird, weicht aufgrund mechanischer Ungenauigkeiten oft von der rechnerischen Mittelstellung ab. Dies darf dennoch nicht als Interaktion erkannt werden.

wraps - Bei Kanälen mit dieser Eigenschaft springt der Wert bei Erreichen des maximalen Wertebereichs auf das Minimum um und umgekehrt. Dieses Verhalten ist z. B. bei Interaktionskomponenten zu beobachten, welche die absolute Rotation eines Knopfes bestimmen, ohne die Anzahl der möglichen Rotationen zu begrenzen.

Codewortkanal

Codewortkanäle übertragen im Gegensatz zu den Wertekanälen nicht die Daten eines physikalischen Sensors, sondern eine Sequenz von Codes, die jeweils eine eigene Semantik haben können. Dieser Kanaltyp wird für Geräte verwendet, die selbst die Sensordaten interpretieren und Ereignisse generieren, wie z. B. Sprach- oder Gestenerkenner.

Beispiel 12

Das GesturePendant ist in der Lage, verschiedene Gesten der Hand zu erkennen. Jeder dieser Gesten wird ein Code zugeordnet, der dann übertragen wird, wenn das Gerät die Geste erkennt.

Größe der Codemenge - Es wird zwischen zwei Arten von Codekanälen unterschieden. Denen mit fester Codemenge (z. B. eine Gestenerkennung) und denen mit dynamischer Codemenge (z. B. Spracherkennung). Bei der dynamischen Codemenge können auch Codes erkannt und übertragen werden, die nicht in der Codeliste auftauchen. Bei einer festen Codemenge hingegen kann sich die Strategie darauf verlassen, dass nur genau diese Codes übertragen werden.

Codemenge - Liste mit bekannten Codes. Jedem Code können zusätzlich eine oder mehrere Semantiken zugeordnet werden. Eine „hoch“ Geste kann also mit der entsprechenden Bedeutung verknüpft werden. Bei fester Codemenge können keine Codes übertragen werden, die nicht in dieser Liste enthalten sind. Bei dynamischer Codemenge kann hier trotzdem einzelnen Codes eine Semantik zugewiesen werden.

Audiokanal

Audiokanäle transportieren Tondaten. Mikrofone dienen der Toneingabe, Lautsprecher der Tonausgabe. Für jeden dieser Kanäle werden die folgenden Eigenschaften gespeichert:

Beispiel 13

Ein Bluetooth-Headset hat sowohl einen Audio-Eingabe- als auch einen Audio-Ausgabekanal. Beide sind in der Regel von niedriger Qualität (8 Bit, 8000 Hz) und mono.

Frequenz - Abtastfrequenz des Datenstroms. z. B. 44100 Hz bei einer Audio CD.

Auflösung - Bitanzahl, mit der ein einzelnes Sample aufgelöst ist. Typischerweise 8 oder 16 Bit.

Anzahl Kanäle - Stereoströme haben zwei Kanäle, Monokanäle hingegen nur einen.

wahrgenommene Lautheit (neu)- Wahrgenommene Lautheit in sone bei maximaler Aussteuerung.

Grafikkanal

Grafikkanäle beschreiben Bitmap-Ausgabegeräte, also Bildschirme oder Head-Mounted-Displays. Die Eigenschaften dienen dazu, einer Interaktionsstrategie Informationen darüber zur Verfügung zu stellen, wie viele Daten auf diesem Bildschirm für einen Menschen sinnvoll dargestellt werden können.

Beispiel 14

Sowohl ein PDA als auch ein HMD haben einen Grafikkanal. Obwohl die physische Größe der beiden Anzeigen sehr unterschiedlich ist, kann die wahrgenommene Auflösung identisch sein. Bei einem HMD erscheinen die Pixel so groß wie bei einem PDA, da der Abstand zum Auge geringer ist.

Breite/Höhe/Farbanzahl - Diese drei Eigenschaften zusammen beschreiben die Fähigkeiten, des mithilfe dieses Kanals angesprochenen, Bildgerätes. Ein HMD der Firma Microoptical hat beispielsweise eine Auflösung von 800 mal 600 Bildpunkten bei 16 Bit Farbauflösung.

Wahrgenommene Pixelgröße (neu) - Raumwinkel, den ein Pixel aus Sicht des Betrachters in einer für das Gerät üblichen Betrachtungsentfernung einnimmt. Dadurch kann die tatsächlich nutzbare Auflösung bestimmt werden.

Kanalgruppe

Zusätzlich können Datenkanäle in Gruppen organisiert werden, um weitere Beziehungen zwischen ihnen zu beschreiben. Jeder Kanal kann dabei Teil mehrerer Gruppen sein. Eine Gruppe wiederum kann neben Datenkanälen auch andere Gruppen enthalten. Es gibt vier verschiedene Typen von Gruppen, die unterschiedliche Beziehungen darstellen:

fusioniert - Gruppiert mehrere Kanäle, die inhärent miteinander verbunden sind. Also Kanäle, die nicht getrennt voneinander beeinflusst werden können. Die Achsen einer Maus sind beispielsweise untrennbar miteinander verbunden, da es nahezu unmöglich ist, die Maus nur genau entlang einer dieser Achsen zu bewegen. Es ist daher nicht sinnvoll, die Kanäle gleichzeitig für unterschiedliche Aufgaben zu verwenden.

angeordnet - Diese Art von Gruppe zeigt an, dass die Kanäle physisch auf eine bestimmte Weise angeordnet sind. Wie genau diese Anordnung aussieht, wird durch die Semantik bestimmt, die der Gruppe zugeordnet werden kann. Ist keine besondere Semantik angegeben, wird angenommen, dass die Kanäle örtlich nahe Aktoren oder Sensoren beschreiben. Bspw. können die Knöpfe einer Zehnertastatur in einer Anordnungsgruppe zusammengefasst und mit der *Semantik*-Eigenschaft „Zehnertastatur“ versehen werden, um die Anordnung der Zahlen gegenüber der Anordnung auf einem Telefon abzugrenzen.

orthogonal - Von allen Kanälen dieser Gruppe kann immer nur ein Kanal aktiv sein, da seine Verwendung die gleichzeitige Verwendung der anderen Kanäle

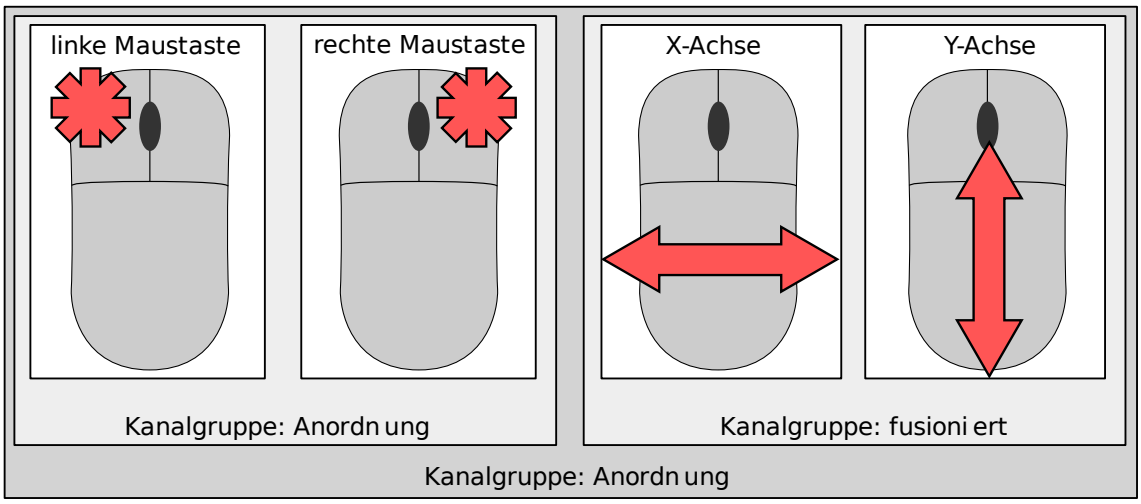

Abbildung 4.5: Die vier Datenkanäle einer Computermaus mit Hilfe von Kanalgruppen zusammengefasst.

ausschließt. Die gegenüberliegenden Knöpfe eines 4-Wege-Joysticks oder Steuerkreuzes weisen zum Beispiel dieses Verhalten auf. Solch ein Steuerkreuz kann entweder nach links oder nach rechts gedrückt werden, nicht jedoch in beide Richtungen gleichzeitig.

logisch - Dieser Typ wird verwendet, um Zusammenhänge zwischen Kanälen zu beschreiben, die durch die anderen Typen nicht abgedeckt werden. Bei einem WiiRemote Controller beispielsweise ist der A-Knopf auf der Oberseite angebracht und wird mit dem Daumen bedient. Der B-Knopf hingegen ist auf der Unterseite und wird mit dem Zeigefinger ausgelöst. Die Knöpfe sind weder fusioniert, noch orthogonal oder in einer bestimmten Anordnung. Allerdings deutet die Beschriftung A/B eine Gruppe der Buchstabenknöpfe an, im Gegensatz zu den anderen Knöpfen des Controllers, die mit +/- und 1/2 bezeichnet sind. Um diesen Zusammenhang darzustellen, kann eine *logische* Gruppe verwendet werden.

Beispiel 15

Aufgrund der Eindimensionalität der Datenkanäle wird eine Computermaus durch vier Wertekanäle beschrieben. Je einen für die beiden Maustasten und je einen für die beiden Bewegungsachsen. Abbildung 4.5 zeigt, wie die vier Kanäle mithilfe von Kanalgruppen wieder in Zusammenhang gebracht werden können. Die Verwendung der beiden Anordnungsgruppen ist durch die physische Nähe ihrer Elemente bestimmt und wird deshalb nicht weiter durch eine Semantik-*Eigenschaft spezifiziert.*

Neben dem Typ kann jeder Gruppe auch eine Semantik zugeordnet werden, welche die Beziehung näher beschreibt:

Beispiel 16

*Die 12 Tasten (0-9, *, #) eines Mobiltelefons können mit einer „angeordnet“ Gruppe und der zusätzlichen* Semantik-*Eigenschaft „Telefontastatur“ gekennzeichnet werden. Eine Interaktionsstrategie zur Texteingabe wie z. B. T9, die genau dieses Layout voraussetzt, kann dann auf diese Knopfgruppe angewendet werden.*

Beispiel: GesturePendant

Um das Zusammenspiel von Benutzermodell und Interaktionsgerätemodell zu demonstrieren, wird die Modellierung des in Kapitel 2.2 beschriebenen GesturePendant erläutert.

Das Gerätemodell enthält zunächst zwei Ressourcenprofile. Eines der Profile beschreibt den passiven Ressourcenbedarf des Geräts. Während der Trageriemen die Rückseite des Nackens bedeckt, befindet sich das GesturePendant in der Mitte der Brust. Das zweite Ressourcenprofil beschreibt die Durchführung der Gesten. Dazu muss die linke Hand vor die Brust bewegt werden, während die Kamera nicht verdeckt sein darf:

```
<device name="GesturePendant">
  <wearing description="Wearing GesturePendant">
    <requirements name="cover">
      <properties key="where" value="body.neck.back"/>
      <properties key="object" value="GesturePendant Strap"/>
    </requirements>
    <requirements name="cover">
      <properties key="where" value="body.chest.front"/>
      <properties key="object" value="GesturePendant"/>
    </requirements>
  </wearing>
  <using description="left hand gesture">
    <requirements name="mustNotBeCovered">
      <properties key="where" value="body.chest.front"/>
      <properties key="object" value="GesturePendant"/>
    </requirements>
    <requirements name="move">
      <properties key="where" value="body.left.arm"/>
      <properties key="target" value="body.chest.front"/>
    </requirements>
  </using>
```

Das GesturePendant erkennt zwei unterschiedliche Typen von Gesten: Kontinuierliche Gesten und Kommandogesten. Kommandogesten werden durch einen Codewortkanal und die kontinuierlichen Gesten durch einen Wertekanal beschrieben. Beide Kanäle verwenden dasselbe Ressourcenprofil:

```
  <channels type="ValueChannel"
      name="Perform_Continuous_Gesture"
      using="//@library/@content.9/@using.0"
      relative="true" volatile="true">
```

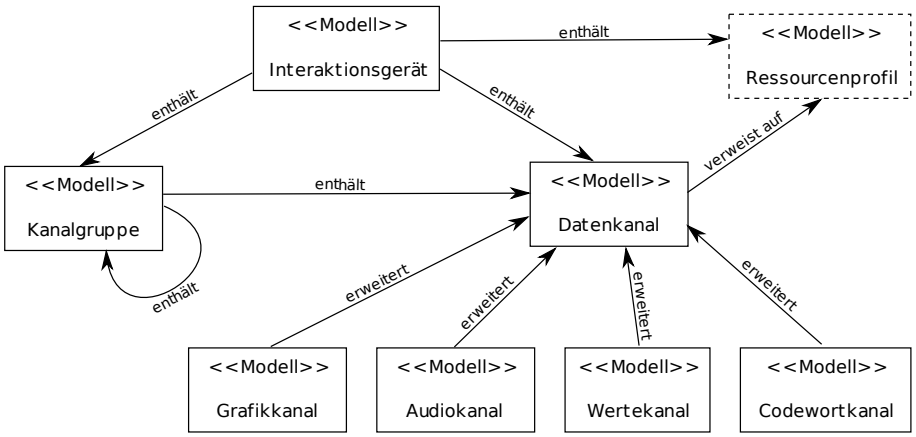

Abbildung 4.6: Beziehungen zwischen Modellelementen des Interaktionsgerätemodells.

```
    <events>StartInteraction</events>
    <events>EndInteraction</events>
    <valueDomain max="100"/>
  </channels>
  <channels type="CodeChannel"
      name="Perform_Command_Gesture"
      using="//@library/@content.9/@using.0">
    <events>StartInteraction</events>
    <events>EndInteraction</events>
    <codes value="gesture 1"/>
    <codes value="gesture 2"/>
    ...
  </channels>
```

Zuletzt wird die gleichzeitige Benutzung von Kommandogesten und kontinuierlichen Gesten durch eine orthogonale Kanalgruppe ausgeschlossen:

```
  <groups name="Can only use one at a time"
      channels="Perform_Command_Gesture Perform_Continuous_Gesture"
      type="orthogonal"/>
</device>
```

Implementierung

Die Umsetzung der Konzepte des Interaktionsgerätemodells wird in Abbildung 4.6 verdeutlicht. Ein Interaktionsgerät enthält sowohl Datenkanäle verschiedenen Typs als auch Gruppen. Eine Gruppe kann wiederum Datenkanäle oder Gruppen enthalten. Die Benutzeranforderungen eines Interaktionsgerätes werden durch Einbindung verschiedener Ressourcenprofile beschrieben. Diese Ressourcenprofile werden direkt im Interaktionsgerät gespeichert. Datenkanäle, die nur unter Verwendung eines bestimmten Ressourcenprofils zur Verfügung stehen, können zusätzlich mit diesem verknüpft werden.

4.4.2 Interaktionsstrategien

Interaktionsstrategien sind das Bindeglied zwischen den in der Arbeitssituation modellierten Mikro-Computeraufgaben und den Datenkanälen der zur Verfügung stehenden Interaktionsgeräte. Jede *Interaktionsstrategie* implementiert genau eine *generische Computeraufgabe*. Um eine Makro-Computeraufgabe umzusetzen, werden deshalb mehrere Interaktionsstrategien benötigt. Interaktionsstrategien sind also eine konkrete Herangehensweise an ein bestimmtes Interaktionsproblem, eine generische Computeraufgabe. Die Forschungsgemeinde im Bereich Mensch-Computer-Interaktion hat im Laufe der Zeit viele dieser Herangehensweisen entwickelt und untersucht (siehe bspw. Konferenzbände des *ACM Symposium on User Interface Software and Technology*). Interaktionsstrategien sollen es dem Designer ermöglichen dieses Wissen zu nutzen, ohne selbst Experte auf diesem Gebiet sein zu müssen.

Die Granularität der Interaktionsstrategien hängt von den definierten generischen Computeraufgaben ab. Sie können sehr grob sein, wie zum Beispiel das auf heutigen Bürocomputern vorherrschende WIMP-Prinzip (Windows, Icons, Menus, Pointer), das ebenfalls als Interaktionsstrategie bezeichnet werden kann. Andererseits kann der Umfang einer Interaktionsstrategie auch sehr begrenzt sein. Zum Beispiel existieren verschiedene Strategien, um in WIMP-Systemen eine Auswahl aus mehreren Elementen zu implementieren (z. B. DropDown-Box, Listbox und Radio-Buttons). Neben diesen Strategien, die alle mit denselben Eingabegeräten arbeiten (Maus und Tastatur) gibt es auch Interaktionsstrategien, die andere Interaktionsgeräte verwenden, um eine Computeraufgabe zu implementieren. Das Konzept der Interaktionsstrategie ist mächtig genug um all diese Fälle abzudecken. Da der Fokus der vorliegenden Arbeit jedoch auf der Integration verschiedenster Interaktionsgeräte liegt, werden auch die Interaktionsstrategien mit einem solchen Fokus beschrieben, wie das folgende Beispiel zeigt:

Beispiel 17

Die Bewegung eines Zeigers auf einem Bildschirm kann beispielsweise mit einem Touchpad, einer Maus oder einem Handy mit Kamera realisiert werden. Ein Touchpad liefert eine absolute Position, welche direkt in eine Zeigerposition umgewandelt werden kann. Die Maus liefert hingegen relative Positionsdaten. Eine entsprechende Interaktionsstrategie muss diese aufaddieren, um eine absolute Zeigerposition zu erhalten. Das Handy wiederum kann aufeinanderfolgende Kamerabilder vergleichen und so die relative Bewegung des Handys bestimmen [JOMS06]. Diese kann dann in eine Zeigerposition umgewandelt werden. Dies sind drei mögliche Strategien für die generische Computeraufgabe „Zeiger positionieren“, die unterschiedliche Interaktionsgeräte verwenden.

Für eine bestimmte generische Computeraufgabe, wie z. B. „Auswahl eines Elements aus einer Liste“(1-aus-N), können also mehrere Strategien mit unterschiedlichen Anforderungen an Datenkanäle existieren. Interaktionsstrategien können sich jedoch nicht nur in der Art und Anzahl der benötigten Datenkanäle unterscheiden, sondern auch in ihrer Eignung für verschiedene Ausprägungen einer konkreten Computeraufgabe.

Beispiel 18

Die Anzahl der zur Auswahl stehenden Elemente bei einer Auswahl 1-aus-N hat einen erheblichen Einfluss darauf, welche Strategie geeignet ist. Bei kleinem N können die Einträge direkt einer entsprechenden Anzahl Knöpfe zugewiesen werden. Bei N=5 also z. B. den fünf Richtungen eines 5-Wege-Joysticks [VVP07]. Bei größeren N hingegen werden allgemeinere Verfahren benötigt, die mit beliebig langen Listen umgehen können. Dies kann zum Beispiel durch Blättern oder Scrollen geschehen.

Insgesamt gibt es eine Vielzahl dieser Kriterien zu berücksichtigen, bevor die „beste“ Interaktionsstrategie bestimmt werden kann. Da eine automatische Auswahl an Interaktionsstrategien jedoch nicht der Fokus dieser Arbeit war, wurde von einer Modellierung dieser Kriterien abgesehen. Sie stehen dem Designer daher nur als beschreibender Text zur Verfügung. Im Folgenden werden die beiden Schnittstellen einer Interaktionsstrategie, die generische Computeraufgabe auf der einen und die Geräteanforderungen auf der anderen Seite, näher erläutert.

Generische Computeraufgaben

Generische Computeraufgaben können sehr unterschiedlich sein. Ihre Granularität hängt von den Anforderungen der Anwendung ab, ebenso wie die konkrete Funktionalität einer solchen generischen Computeraufgabe. Ansätze modellbasierter Schnittstellenbeschreibung wie UsiXML [LV04] oder XForms [Con07] verwenden abstrakte Aufgabenbeschreibungen, um von der konkreten Hardware und Software Plattform unabhängig zu sein. Dabei beschränken sie sich jedoch auf eine feste Liste weniger generischer Computeraufgaben, um die Entwicklung von Laufzeitumgebungen zu vereinfachen. Dieses Konzept ist für formularbasierte Anwendungen ausreichend, andere interaktivere Anwendungsarten sind hier jedoch nicht vorgesehen. Die Interaktion mit einer Landkarte, die Zoomen und Verschieben erlauben soll, kann mit diesen Verfahren beispielsweise nur schwer modelliert werden.

Um dieses Problem zu vermeiden, wurde hier ein anderer Ansatz gewählt. Anstatt die Liste der generischen Computeraufgaben fest vorzugeben, können hier beliebig viele generische Computeraufgaben in einer Bibliothek zur Verfügung gestellt werden. Eine solche generische Computeraufgabe besteht dann aus einer natürlichsprachlichen Beschreibung für den Designer und einer Schnittstellenbeschreibung. Diese spezifiziert genau, welche Informationen von Anwendung und Strategie ausgetauscht werden. Mithilfe dieser Schnittstellenbeschreibung können nun verschiedene Strategien definiert werden, die diese Computeraufgabe umsetzen. Durch diese Art der Aufgabenbeschreibung ist die Granularität der Computeraufgaben völlig offen. Eine existierende grafische Anwendung kann z. B. Computeraufgaben für Texteingabe und Zeigerbewegung definieren und ansonsten die existierenden Mechanismen des Betriebssystems verwenden. Eine andere Anwendung kann hingegen sehr spezifische Computeraufgaben verwenden, um Teilprobleme mit speziellen Interaktionsgeräten zu lösen.

Im Folgenden wird beispielhaft eine generische Computeraufgabe „Navigation“ beschrieben, die sowohl im Endoskopieszenario, als auch in den beiden, in Abschnitt 2.3 vorgestellten Szenarien, Verwendung findet.

Beispiel 19

Die generische Computeraufgabe „Navigation“ *ermöglicht es dem Benutzer, innerhalb einer Anwendung oder eines Datenbestandes zu navigieren. Die generische Computeraufgabe liefert der Anwendung die folgenden Ereignisse:* vorwärts, rückwärts, auswählen *und* abbrechen. *Ein Datenaustausch von der Anwendung zur Interaktionsstrategie findet nicht statt.*

Die Aufgabe erfüllt die Definitionen 7 und 8. Die einzelnen Teilaspekte der generischen Computeraufgabe lassen sich nicht mehr sinnvoll voneinander trennen. Sie lässt sich aber so implementieren, dass nur jeweils eine einzige Interaktion des Benutzers mit dem Systen notwendig ist, beispielsweise mit vier Knöpfen. Durch die semantische Beziehung der vier Teilfunktionen ist diese generische Computeraufgabe jedoch mehr, als nur die Summe von vier generischen Aktionen.

Die Aufgabe ist außerdem generisch genug, um in vielen verschiedenen Szenarien einsetzbar zu sein. Sowohl im Endoskopieszenario, als auch im Visitenszenario kann sie verwendet werden, um durch die verfügbare Dokumentation zu navigieren. Im Szenario Automobilproduktion hingegen lässt sie sich verwenden, um die Einsicht der Hilfe und Dokumentation zu modellieren.

Geräteanforderungen

Mithilfe der generischen Computeraufgaben kann zwar die Funktion einer Strategie beschrieben werden, welche Interaktionsgeräte für die Umsetzung benötigt werden muss jedoch ebenfalls spezifiziert werden. Um nicht von konkreten Interaktionsgeräten abhängig zu sein, wird das Interaktionsgerätemodell für diese Beschreibung verwendet. Interaktionsstrategien formulieren Anforderungen an Datenkanäle und deren Eigenschaften. Da häufig nicht nur ein einzelner Datenkanal benötigt wird, müssen dabei auch die Beziehungen zwischen den verschiedenen Datenkanälen berücksichtigt werden.

Beispiel 20

Eine Strategie zur Umsetzung der generischen Computeraufgabe aus Beispiel 19 kann z. B. vier Knöpfe für die vier Funktionen verwenden. Ein Knopf wird dabei als ein Wertekanal beschrieben, der einen Wertebereich von genau 2 Zuständen hat, wobei einer dieser Zustände bevorzugt ist (vgl. 4.4.1). Zusätzlich können für die Paare „vor/zurück“ und „auswählen/abbrechen“ solche Knöpfe präferiert werden, die örtlich nah sind, also z. B. in einer Anordnungsgruppe. Auf keinen Fall dürfen zwei der ausgewählten Knöpfe in einer fusioniert-Beziehung stehen, da sie unabhängig voneinander betätigt werden müssen. Des Weiteren kann auf die Semantik der Knöpfe eingegangen werden. Die Funktionen „vor“ und „zurück“ werden beispielsweise häufig mit einer links/rechts-Bewegung in Verbindung gebracht, sodass Knöpfe, die bereits diese Semantik haben, bevorzugt werden können.

Da die Anforderungen an die Datenkanäle nicht immer einer einfachen Abfrage der Eigenschaften entsprechen, sondern auch voneinander und von den Vorgaben der Computeraufgabe abhängig sein können, müsste eine abstrakte Abfragesprache sehr mächtig sein. Da eine solche Abfragesprache nicht das Ziel dieser Arbeit

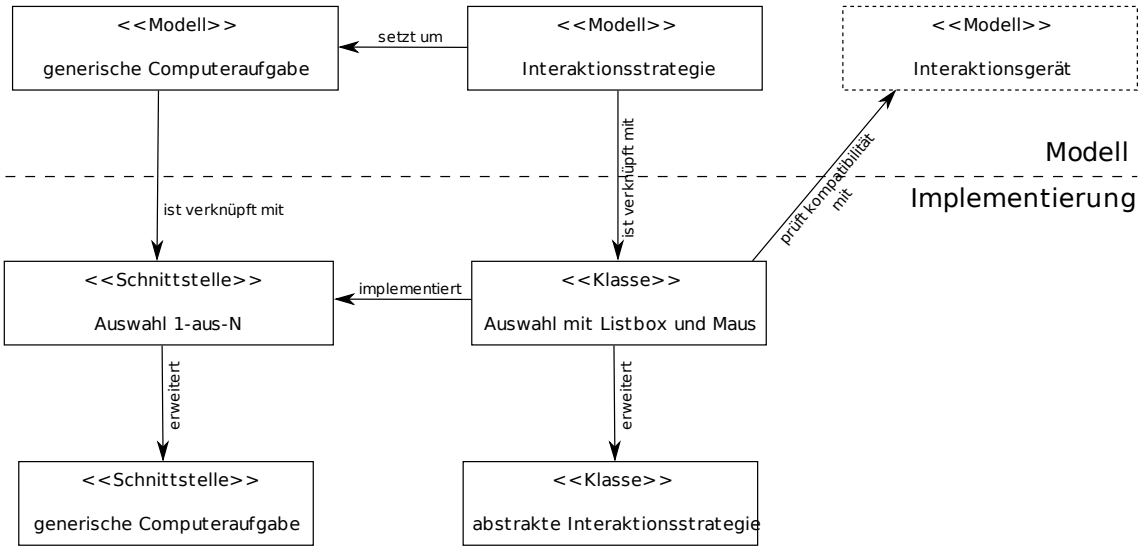

Abbildung 4.7: Kopplung zwischen Modell und Implementierung bei Interaktionsstrategien und generischen Computeraufgaben.

war, wurde ein pragmatischer Ansatz gewählt. Die Selektion geeigneter Datenkanäle findet innerhalb einer Interaktionsstrategie statt und wird für jede Interaktionsstrategie gesondert implementiert. Dazu erhält die Interaktionsstrategie eine Liste mit verfügbaren Datenkanälen. Diese werden dann analysiert und eine List kompatibler Konfigurationen wird zurückgeliefert. Dabei kann die Interaktionsstrategie auf eine Reihe von Hilfsfunktionen zurückgreifen, welche die Suche nach bestimmten Datenkanälen vereinfachen.

Implementierung

Die generischen Computeraufgaben und Interaktionsstrategien sind wie in Abbildung 4.7 dargestellt implementiert. Generische Computeraufgaben werden durch zwei Komponenten spezifiziert. Ein Java-Interface, das die notwendige Schnittstelle für die Anwendung definiert und ein Modellelement, das einen Bezeichner sowie eine natürlichsprachliche Beschreibung enthält. Das Modellelement der generische Computeraufgabe ist der Teil, der vom Designer verwendet wird, um die Makro-Computeraufgaben einer Arbeitssituation zu beschreiben. Das Interface hingegen ist notwendig, um dem Entwickler später die Implementierung verschiedener Interaktionsstrategien zu ermöglichen.

Eine ähnliche Teilung findet auch bei den Interaktionsstrategien statt. Das Modellelement Interaktionsstrategie enthält einen Bezeichner, eine natürlichsprachliche Beschreibung und den Bezeichner der umgesetzten generischen Computeraufgabe. Auf der Implementierungsebene ist eine Interaktionsstrategie eine Java-Klasse, die von einer abstrakten Interaktionsstrategieklasse erbt und das Interface der umgesetzten generischen Computeraufgabe implementiert. Die Klasse von der eine konkrete Interaktionsstrategie erbt, definiert z. B. Funktionen um aus einer Menge an Interaktionsgeräten eine geeignete Konfiguration an Datenkanälen für die Umsetzung der Interaktionsstrategie zu ermitteln. Diese wird von einer konkreten Interaktionsstra-

tegie dann mit spezifischen Code gefüllt, der Kriterien für geeignete Datenkanäle festlegt.

4.4.3 Bibliotheken

Um Designwissen aus vorherigen Projekten einfach zugänglich zu machen, werden einmal erstellte Komponenten des Computersystems in einer Bibliothek gespeichert. Diese Bibliothek enthält dann Instanzen der jeweiligen Modelle. Solche Bibliotheken gibt es für generische Computeraufgaben, Interaktionsstrategien und Interaktionsgeräte. Die Wiederverwendung von generischen Computeraufgaben ist deshalb so wichtig, weil mit ihnen auch die Interaktionsstrategien wiederverwendet werden können. Der Designer wird dadurch außerdem in die Lage versetzt Methoden für seinen Entwurf zu berücksichtigen, die er vorher nicht kannte. Dasselbe gilt für Interaktionsgeräte, die ebenfalls automatisch auf ihre Kompatibilität mit der Arbeitssituation untersucht werden können.

4.4.4 Zusammenfassung

Das Computersystemmodell besteht aus zwei Teilen, dem *Gerätemodell* und den *Interaktionsstrategien*. Beide Teile zusammen ermöglichen es, geeignete Interaktionsgeräte für die Implementierung einer Computeraufgabe des Benutzers zu identifizieren.

Das Gerätemodell beschreibt ein *Interaktionsgerät* dabei als Sammlung von *Kanälen*, die jeweils durch *Eigenschaften* näher beschrieben werden. Dabei wurden neben den bekannten Eigenschaften aus der Literatur auch neue Eigenschaften definiert, welche die Berücksichtigung Wearable-Computing-spezifischer Probleme ermöglichen. So können die *Anforderungen eines Interaktionsgerätes an den Benutzer* erstmals durch *Ressourcenprofile* modelliert werden. Damit wird es möglich automatisch herauszufinden, ob ein Interaktionsgerät parallel zu einer bestimmten Arbeitssituation eingesetzt werden kann. Zusätzlich ermöglichen *Gruppen*, die Beziehungen mehrerer Kanäle untereinander näher zu bestimmen.

Des weiteren wurde das Konzept der *Interaktionsstrategien* vorgestellt, das die Verknüpfung von Interaktionsgeräten und *Computeraufgaben* des Benutzers ermöglicht. Eine Interaktionsstrategie implementiert dabei jeweils eine *generische Computeraufgabe*. Die Anforderungen an die dafür verwendeten Interaktionsgeräte sind abstrakt mithilfe der *Gerätebeschreibung* formuliert, sodass eine Interaktionsstrategie unabhängig von konkreten Interaktionsgeräten verwendet werden kann.

Sowohl Gerätemodelle als auch Interaktionsstrategien werden in *Bibliotheken* gespeichert, die es dem Designer ermöglichen auf *bestehendes Designwissen* zurückzugreifen. Durch die Entkopplung des Computersystemmodells vom Arbeitssituationsmodell wird die Wiederverwendbarkeit der Modelle dabei unterstützt.

Die gerade beschriebenen Modelle dienen als Grundlage für die Unterstützung des in Kapitel 3 vorgestellten Entwurfsprozesses. Wie eingangs erwähnt, werden nun die Algorithmen und prototypischen Werkzeuge im Rahmen der jeweiligen Prozessschritte näher erläutert.

4.5 Prozess: Datenerhebung

Da der Prozessschritt der Datenerhebung von der Durchführung verschiedener Methoden der Feldforschung dominiert wird, ist die Unterstützung vor allem durch die Prozessbeschreibungen und Arbeitsvorschriften in Kapitel 3 geprägt. Die Durchführung einer computergestützten Sequenzanalyse wird jedoch durch den TaskObserver ermöglicht. Diese spezielle Anwendung für mobile Beobachtungen ohne Videoaufzeichnung wird nun detailliert beschrieben.

4.5.1 TaskObserver

Das TaskObserver Werkzeug ermöglicht die computerunterstützte Sequenzanalyse von Arbeitssituationen. Ziel der Entwicklung war es ein Werkzeug zu schaffen, das auch in solchen Umgebungen einsetzbar ist, in denen keine Videoaufzeichnungen erlaubt sind. Unter dieser Bedingung wurde versucht die Datenqualität der Beobachtungen zu optimieren, ohne wesentlich mehr Zeit in Anspruch zu nehmen als für die eigentliche Beobachtung benötigt wird. Um dies zu erreichen, wurde darauf geachtet, dass die Bedienung des Werkzeugs den Beobachter möglichst wenig ablenkt. Außerdem können die so gewonnenen Daten direkt zur Korrektur oder Weiterverarbeitung in die Werkzeuge ProjectEditor und WearableDesigner übernommen werden. In diesem Abschnitt wird zunächst auf den Entwurf der TaskObserver Anwendung eingegangen. Anschließend wird erläutert, wie die Anwendung möglichst effektiv eingesetzt werden kann, um den in Kapitel 3.2 beschriebenen Prozess zu unterstützen.

Design

Um die Bedienung des TaskObserver so intuitiv wie möglich zu gestalten, wurde eine Implementierung auf Basis eines TabletPC gewählt, da die Bedienung mit einem Stift einfach und schnell ist. Abbildung 4.8 zeigt den prinzipiellen Aufbau der Oberfläche und Abbildung 4.9 das Werkzeug auf einem TabletPC. Die zu beobachtenden Ereignisse sind hierbei in einem Raster angeordnet, das den überwiegenden Teil des Bildschirms einnimmt. Wird ein Ereignis beobachtet, dann wählt der Benutzer den entsprechenden Knopf aus und ein grüner Rahmen um das Ereignis zeigt an, dass das Ereignis nun aktiv ist. Durch erneutes Auswählen wird das Ereignis beendet und der grüne Rahmen verschwindet. Zusätzlich ist jedes Ereignis mit einer farblich kodierten Kategorie versehen, die das Auffinden und Erlernen bestimmter Ereignisse erleichtern soll.

Im Vergleich zu einer Listendarstellung, bei der die Ereignisse untereinander aufgeführt werden, sind die Auswahlfelder bei der gewählten Rasterdarstellung, sowohl in horizontaler als auch in vertikaler Richtung groß, was die Zeit, die für eine Auswahl benötigt wird, gemäß Fitt's Law [AZ03] minimiert.

Für den Fall, dass während einer Beobachtung Ereignisse auftreten, die nicht bereits Teil der initialen Ereignisliste sind, ist außerdem die Möglichkeit gegeben neue Knöpfe hinzuzufügen. Zu diesem Zweck existiert ein zusätzliches Feld, bestehend aus einem Knopf und einer Zeichenfläche. Der Ablauf ist dabei wie folgt (siehe auch Abb. 4.10):

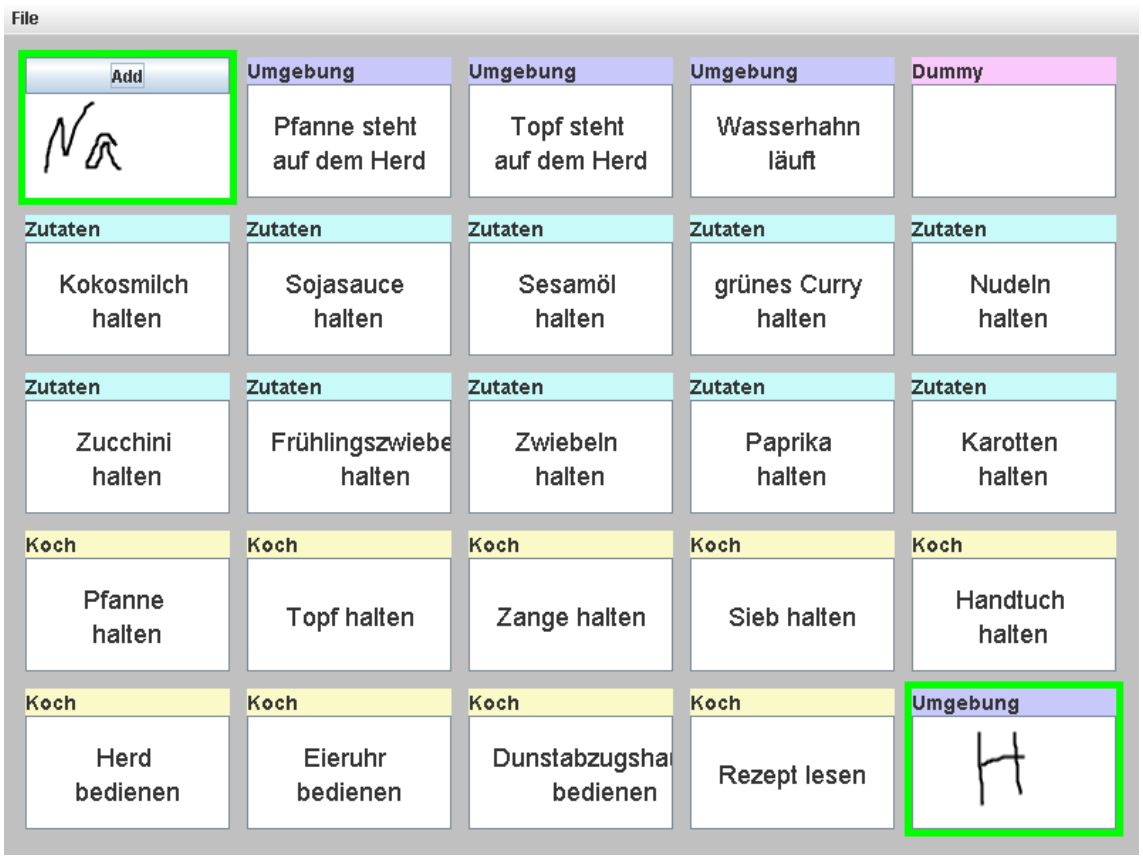

Abbildung 4.8: Oberfläche des TaskObserver. Jeder Kasten ist einem zu beobachtenden Ereignis zugeordnet. Farblich unterschiedene Kategorien oberhalb des Ereignisnamens erleichtern das Auffinden von Ereignissen. Ein grüner Rahmen zeigt an, dass das Ereignis gerade stattfindet. Der Kasten in der linken oberen Ecke wird zum Neuanlegen von Ereignissen verwendet. Neue Ereignisse erhalten eine voreingestellte Kategorie. Die Anordnung der Ereignisse kann frei gewählt werden.

1. Zunächst wird der Knopf gedrückt, um die Aufzeichnung des Ereignisses zu starten.

2. Dann wird die Zeichenfläche verwendet, um ein Symbol für das Ereignis zu erstellen.

3. Der Knopf wird erneut gedrückt, um das Ereignis in das Raster mit den anderen Knöpfen zu übernehmen.

4. Nun kann das Ereignis ganz normal beendet werden.

Im Gegensatz zu den initialen Ereignissen, wird für neue Ereignisse eine Grafik als Bezeichner verwendet. Diese werden gespeichert, um später eine Zuordnung und Benennung der Ereignisse zu ermöglichen. Die Position des Feldes für neue Ereignisse richtet sich danach, ob der Beobachter Rechts- oder Linkshänder ist. Für Rechtshänder befindet sich das Feld in der linken oberen Ecke, damit das Handgelenk beim Schreiben aufgelegt werden kann. Für Linkshänder befindet sich das Feld

Abbildung 4.9: TaskObserver auf einem TabletPC.

dementsprechend in der rechten oberen Ecke. Die Möglichkeit zunächst die Aufzeichnung zu starten und dann erst den Bezeichner zu vergeben, ist deshalb wichtig, weil die Bedeutung eines Ereignisses dem Beobachter oft nicht gleich klar ist. In solchen Fällen kann sofort mit der Aufzeichnung begonnen werden, während der Bezeichner erst später vergeben wird.

Eine eigens für dieses Werkzeug durchgeführte Benutzerstudie (siehe Kapitel 5.3) hat ergeben, dass eine gute Erlernbarkeit der Ereignispositionen im Raster für eine hohe Datenqualität essenziell ist. Aus diesem Grund wurden zum einen farbliche Markierungen der Ereignisse mithilfe von Kategorien ermöglicht, die das Auffinden erleichtern. Zum anderen können die Positionen der Ereignisse vor Beginn der Beobachtung definiert werden. Auf diesem Weg ist es z. B. möglich aufeinanderfolgende Prozessschritte nebeneinander in einer Reihe anzuordnen. Der daraus entstehende Zusammenhang zwischen zeitlicher Abfolge und räumlicher Anordnung erleichtert die Suche nach dem richtigen Ereignis und erhöht somit die Datenqualität.

Außerdem hat die Benutzerstudie ergeben, dass sehr kurze Ereignisse nur schlecht in ihrer Dauer beobachtet werden können. Deshalb gibt es die Möglichkeit solche Ereignisse nicht als Intervall zu erfassen, sondern als diskretes Ereignis ohne zeitliche Ausdehnung. Bei solchen Ereignissen führt die Auswahl dazu, dass ein Intervall mit vordefinierter Länge gespeichert wird. Das entsprechende Feld erhält keinen grünen Rahmen und ist sofort wieder für die nächste Erfassung bereit. Zusätzlich wird angezeigt, wie oft das Ereignis schon erfasst wurde.

Verwandte Arbeiten

Das hier beschriebene Werkzeug basiert auf Ideen, die von Harrison u. A. im Timelines-Werkzeug[HOB94] und von Weber u. A. im Marquee-Werkzeug [WP94] umgesetzt wurden. Timelines ist ein Werkzeug, das zur Kodierung von Videoaufzeichnung verwendet wird. Aus diesem Grund ist es für den mobilen Einsatz nicht geeignet. Jedoch wurden hier bereits Benutzerdefinierbare Knöpfe zur schnellen Eingabe von Kodes verwendet.

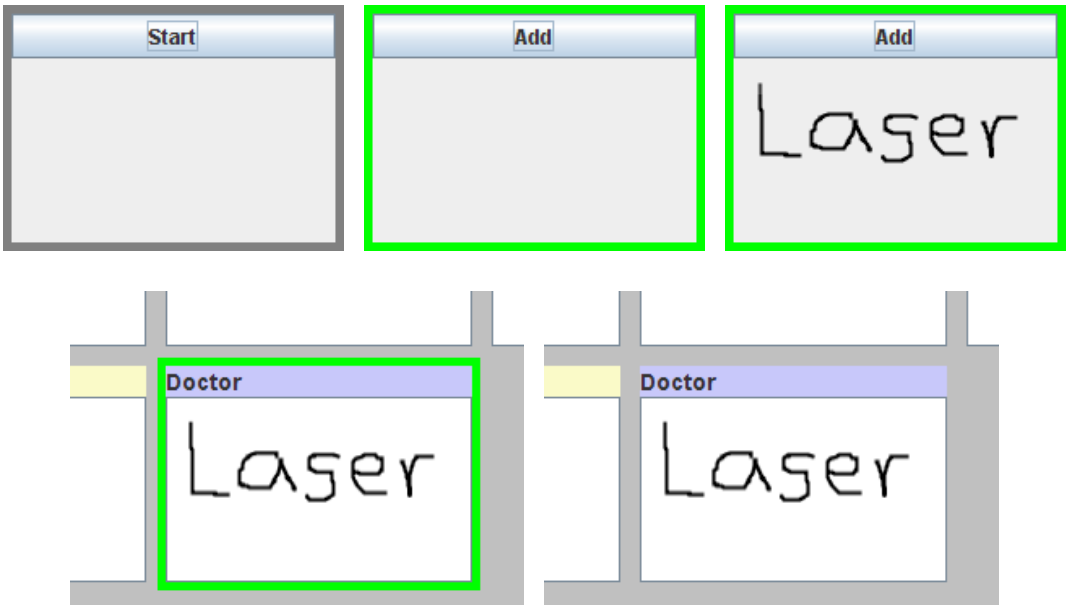

Abbildung 4.10: Sequenz um ein neues Ereignis zu erstellen (l.n.r.,o.n.u.).

Die Kodierung einer Beobachtung in Echtzeit wird durch das Werkzeug Marquee ermöglicht. Hier wird ein Stifteingabe verwendet um Stichwörter zu definieren, die im weiteren Verlauf als Benutzerdefinierte Knöpfe verwendet werden. Zusätzlich kann der Benutzer Zeitzonen definieren und in diesen Notizen machen. Der Fokus dieses Werkzeugs liegt jedoch auf der Erstellung von Notizen und nicht auf der zeitgenauen Erfassung von Ereignissen, weshalb das Werkzeug für die hier beschreibene Anwendung nicht geeignet ist.

Verfügbare kommerzielle Anwendungen wie INTERACT [man08] oder The Observer [nol07] bieten zwar ähnliche Funktionen an, setzen aber bei schnellen Abfolgen von Ereignissen auf parallele Videoaufzeichnungen und andere Formen der automatischen Datenerfassung (z. B. mit Sensoren). Manuelle Annotationen in Echtzeit sind zwar möglich, die mobilen Werkzeuge sind jedoch nicht auf den vorliegenden Anwendungsfall hin optimiert und deshalb nur begrenzt geeignet.

Initiale Ereignisliste

Bevor der TaskObserver für eine Beobachtung eingesetzt werden kann, muss eine initiale Ereignisliste erstellt werden. Die richtige Auswahl der zu beobachtenden Ereignisse ist dabei entscheidend für den gesamten weiteren Prozess und die Qualität der Beobachtungen (siehe Abschnitt 5.3). Sind die Ereignisse zu grob ausgewählt, lässt sich später keine sinnvolle Analyse durchführen. Werden zu feine Ereignisse ausgewählt, steigt der Aufwand drastisch an und Beobachtungen werden zunehmend ungenauer. Die richtige Granularität der Ereignisse hängt also sowohl von der späteren Analyse ab, als auch von der Methode, die zur Beobachtung verwendet wird.

Bei dem in Kapitel 3 beschriebenen Prozess können für die initiale Aufgabenliste vorläufige Ergebnisse der Aufgabenstruktur verwendet werden, die durch vorange-

gangene Interviews oder Beobachtungen im Kontext gewonnen wurden.

Bei der Auswahl der richtigen Knoten aus der Aufgabenstruktur gilt es folgende Regeln zu berücksichtigen:

1. Die Aufgabe sollte für die Integration eines Computersystems in die Arbeitssituation von Bedeutung sein (siehe Regel 1).

2. Die benötigten Interaktionsressourcen sollten über die Dauer der Aufgabe konstant sein (siehe Regel 1).

3. Bei einer computergestützten Sequenzanalyse gilt außerdem, dass die durchschnittliche Dauer der Aufgabe weder zu lang (ca. $> 20sek$), noch zu kurz (ca. $< 4sek$) sein sollte (siehe Abschnitt 5.3).

Beispiel 21

Bei der Endoskopieuntersuchung können Teile der Vor- und Nachbereitung weggelassen werden, da sie weder Einfluss auf die eigentliche Untersuchung, noch auf Dokumentation und Dokumenteneinsicht haben, wie z.B. das Desinfizieren des Endoskops. Die Bedienung des Endoskops hingegen muss in mehrere Teilaufgaben unterteilt werden, da der Arzt mal nur die linke Hand verwendet und mal beide Hände. Der Ressourcenbedarf ist also nicht konstant. Sehr kurze Ereignisse wie z.B das Anfertigen eines Fotos können nur als diskrete Ereignisse erfasst werden, da sie für die Erfassung als Intervall zu kurz sind.

Zu den Aufgaben kommen schließlich noch die Umgebungseinflüsse hinzu, die sich während der Arbeitssituation verändern. Ist der Lärmpegel z. B. konstant hoch, muss dies nicht gesondert erfasst werden. Wird das Licht während einer Endoskopieuntersuchung jedoch an- und ausgeschaltet, muss dies als Ereignis erfasst werden. Häufig sind solche Ereignisse jedoch deutlich länger als 20 Sekunden. In solchen Fällen kann es sinnvoll sein, das Ereignis durch zwei Ereignisse zu repräsentieren, die jeweils Start und Ende markieren (siehe Abschnitt 5.3).

Beispiel 22

Das Licht wird während der Endoskopieuntersuchung an- und ausgeschaltet. Anstatt das Ereignis „Licht ist aus“ zu beobachten, werden die diskreten Ereignisse „Licht wird ausgeschaltet“ und Licht wird eingeschaltet“ beobachtet. Aus diesen beiden Ereignissen kann dann später das Ereignis „Licht ist aus“ rekonstruiert werden.

4.5.2 Zusammenfassung

Mit dem TaskObserver wird ein Werkzeug zur Verfügung gestellt, das speziell auf die Bedürfnisse eines Beobachters zugeschnitten ist, der Traces für die Verwendung im vorliegenden Entwurfsprozess erhebt. Der Anwendungsfall wird dabei durch eine endliche Anzahl von Ereignissen charakterisiert, die nahezu sekundengenau kodiert werden müssen und gleichzeitig auftreten können. Die Benutzerschnittstelle wurde auf diesen Anwendungsfall hin optimiert. Zusätzlich hat eine Benutzerstudie (siehe Abschnitt 5.3) eine Reihe von Regeln für die Erstellung der initialen Ereignisliste

ergeben, die zur Verbesserung der Datenqualität beitragen. Die so gewonnen Traces können nun im nächsten Prozessschritt, der Modellierung, weiter verarbeitet werden.

4.6 Prozess: Modellierung

Der nächste Prozessschritt ist die Modellierung der Arbeitssituation. Hier werden zwei prototypische Werkzeuge zur Verfügung gestellt, um die in Abschnitt 3.2 beschriebenen Aktivitäten zu unterstützen. Das eine Werkzeug ist der WearableDesigner, der dazu dient, Traces zu bearbeiten und einfache Änderungen am Primäraufgabenmodell durchzuführen. Für alle weiteren Modellierungsarbeiten steht der ProjectEditor zur Verfügung, der die Bearbeitung aller Modelle ermöglicht. Diese beiden Werkzeuge werden im Folgenden vorgestellt.

4.6.1 WearableDesigner

Der WearableDesigner ist ein Werkzeug, das dazu dient, Traces zu visualisieren und zu bearbeiten. Für die Visualisierung wird die bereits aus Beispielen bekannte Zeitleistendarstellung verwendet. Abbildung 4.11 zeigt wie ein Trace mithilfe des Wearable Designers dargestellt wird. Der Beobachter kann die Ereignisse und Primäraufgabenelemente in dieser grafischen Darstellung direkt bearbeiten und so Beobachtungsfehler beheben. Er ist außerdem in der Lage die Primäraufgabenelemente verschiedener Traces zu konsolidieren. Dazu können bei einer Beobachtung neu hinzugekommene Primäraufgabenelemente auch in andere Traces übernommen oder umbenannt werden. Der WearableDesigner erlaubt es zudem künstliche Traces anzulegen und so Szenarien zu modellieren, die nicht beobachtet werden konnten.

Der WearableDesigner ist als Eclipse Plug-In implementiert und nutzt das EMF-Modell der Arbeitssituation. Zur Darstellung wird das GEF (Graphical Editing Framework) [gef07] verwendet. Die Nutzung existierender Standards erlaubt eine bessere Integration mit anderen Werkzeugen zur Entwurfsunterstützung.

4.6.2 ProjectEditor

Der ProjectEditor ist ein Teil der prototypischen Implementierung der Modellierungsumgebung. Da für die Implementierung der hier beschriebenen Modelle das Eclipse Modeling Framework [emf07](EMF) eingesetzt wurde, war es möglich einen generischen Editor automatisch generieren zu lassen. Mit dem ProjectEditor lassen sich all diejenigen Aspekte der Modelle erstellen und bearbeiten, für die kein spezielles Werkzeug entwickelt wurde.

In der Modellierungsphase sind dies insbesondere die Ressourcenprofile, die Primäraufgaben sowie die Makro- und Mikro-Computeraufgaben. Außerdem ist es möglich, die Bibliothek der Interaktionsgeräte und der generischen Computeraufgaben zu bearbeiten. Abbildung 4.12 zeigt, wie mit dem ProjectEditor das Ressourcenprofil einer Primäraufgaben bearbeitet werden kann.

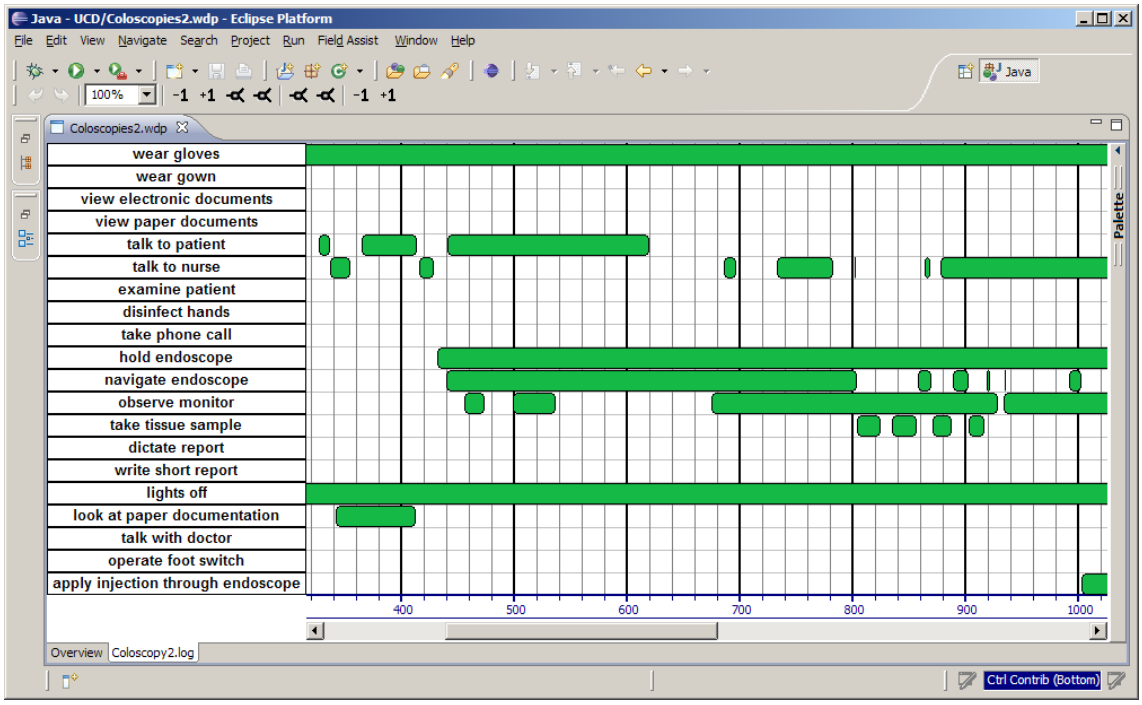

Abbildung 4.11: Der WearableDesigner ist in der Lage Traces zu visualisieren. Er erlaubt außerdem die Bearbeitung und Konsolidierung verschiedener Traces.

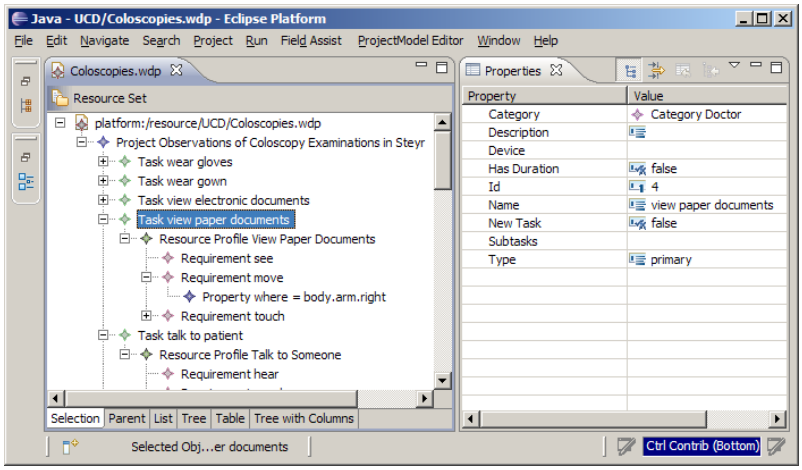

Abbildung 4.12: ProjectEditor Eclipse Plug-In. Die linke Seite zeigt die Modellstruktur. Das Properties-Fenster ermöglicht die Bearbeitung von Eigenschaften einzelner Modellelemente.

4.7 Prozess: Entwurf

Auf die Modellierungsphase folgt der letzte Prozessschritt, der Entwurf. Dieser besteht wie in Abschnitt 3.5 beschreiben aus den vier Teilschritten *Analysieren*, *Visualisieren*, *Entscheiden* und *Modifizieren*, die iterativ ausgeführt werden. Bis auf den Teilschritt *Entscheiden* werden alle Teilschritte direkt durch Werkzeuge unterstützt. Der Entscheidungsprozess ist jedoch ein kreativer Prozess, der nur indirekt durch die anderen drei Teilschritte unterstützt wird. Deshalb wird für diesen Teilschritt kein gesondertes Werkzeug benötigt.

Im Folgenden werden zunächst die Algorithmen beschrieben, welche die automatische Analyse einer Arbeitssituation erlauben. Anschließend werden die Möglichkeiten zur Visualisierung im WearableDesigner beschrieben, sowie die Möglichkeiten zur Modifikation der Modelle.

4.7.1 Analysieren

Der erste Teilschritt, das Analysieren der Arbeitssituation findet vollautomatisch durch das Modell statt. Hier werden die folgenden zwei Analysen durchgeführt, deren Ergebnisse abgespeichert werden, um anschließend für die Visualisierung zur Verfügung zu stehen:

- Kompatibilität von Interaktionsgeräten
- Ermittlung möglicher Interaktionsstrategien

Die *Kompatibilität von Interaktionsgeräten* mit einem Trace berechnet für jedes Interaktionsgerät aus der Bibliothek, ob sein passiver Ressourcenbedarf im Laufe des Traces zu Konflikten führt. Dann ist das Gerät für diesen Trace unbrauchbar. Ist der passive Ressourcenbedarf jedoch kompatibel, wird der aktive Ressourcenbedarf überprüft. Jetzt werden Zeitbereiche berechnet in denen das Interaktionsgerät aktiv verwendet werden kann, um dem Designer Aufschluss über die Situationen zu geben in denen des Interaktionsgerät nutzbar ist. Zu diesem Zweck wird eine *Simulation* der Arbeitssituation durchgeführt. Hat der Designer sich für geeignete Interaktionsgeräte entschieden und diese zu einer Gerätekonfiguration zusammengeschlossen, folgt die *Ermittlung möglicher Interaktionsstrategien*. Hier wird ermittelt ob und mit welchen Interaktionsstrategien sich die definierten Makro-Computeraufgaben implementieren lassen.

Nachfolgend wird zunächst die Vorgehensweise bei der Benutzersimulation beschrieben. Danach folgt der Algorithmus zur Berechnung der Gerätekompatibilität und anschließend das Vorgehen zur Ermittlung möglicher Interaktionsgeräte.

Benutzersimulation

Die Benutzersimulation wird durch das in Abschnitt 4.2 beschriebene Benutzermodell und die dazugehörige Beeinflussungssprache ermöglicht. Das Benutzermodell beschreibt dabei immer den aktuellen Zustand des Benutzers. Durch Hinzufügen

und Entfernen von Ressourcenprofilen, die durch Primäraufgaben und Interaktionsgeräte definiert sind, wird dieser Zustand dann verändert. Eine Simulation läuft deshalb folgendermaßen ab:

1. Instanziierung eines initialen Benutzermodells
2. Hinzufügen/Entfernen von Ressourcenprofilen
3. Konsistenzprüfung
4. Weiter bei 2, bis alle Ereignisse abgearbeitet sind.

Die Simulation externer Einflüsse auf das Benutzermodell wird durch Hinzufügen und Entfernen von Ressourcenprofilen bewerkstelligt. Während der Simulation ist sowohl die Inspektion der gerade verfügbaren Ressourcen möglich, als auch eine Überprüfung der Kompatibilität verschiedener Ressourcenprofile.

Initiales Benutzermodell Zunächst wird ein Benutzermodell im initialen Zustand instanziiert. Im initialen Zustand sind zunächst alle Ressourcen unbenutzt und in jeder Körperregion befindet sich die Haut als einziges Element auf dem Objektstapel. Der simulierte Benutzer ist also nackt.

Hinzufügen/Entfernen von Ressourcenprofilen Ein Ressourcenprofil wird der Simulation hinzugefügt, indem alle Operatoren des Ressourcenprofils in der Reihenfolge ihrer Definition in der Simulation aktiviert werden. Kommt es bei einer dieser Aktivierungen zu einem Konflikt mit dem aktuellen Benutzerzustand, werden bereits hinzugefügte Operatoren wieder entfernt und das Ressourcenprofil wird als inkompatibel klassifiziert.

Ebenso können bereits aktive Ressourcenprofile aus der Simulation entfernt werden, indem die Operatoren in umgekehrter Reihenfolge wieder deaktiviert werden. Hierbei kann es ebenfalls zu Konflikten kommen, wenn z. B. ein Objekt entfernt werden soll, das sich nicht oben auf dem Objektstapel befindet (**cover**-Operator).

Änderungen der aktiven Operatoren, die keine Konflikte verursachen führen zu einem Simulationszustand, der gemäß der verfügbaren Interaktionsressourcen möglich ist. Allerding können Anforderungen verletzt sein, weshalb eine zusätzliche Konsistenzprüfung notwendig ist.

Konsistenzprüfung Die Konsistenz eines Benutzermodells ist folgendermaßen definiert:

Definition 13 (Konsistenz)

Eine Instanz des Benutzermodells ist **konsistent***, wenn alle aktiven Anforderungen erfüllt sind.*

Die Konsistenzprüfung testet ob alle aktiven Anforderungsoperatoren (*mustNotTouch*, *requiresSkinAccess* und *mustNotBeCovered*) eingehalten werden. Da bereits aktive Anforderungen durch das Hinzufügen neuer Operatoren beeinflusst werden können, ist diese Prüfung für jeden aktiven Operator und nach jeder Änderung am Zustand der Simulation notwendig.

Ermittlung der Interaktionsgerätekompatibilität

Im Folgenden wird ein Algorithmus beschrieben, der es auf Basis der Benutzersimulation ermöglicht, Zeitbereiche einer Arbeitssituation zu ermitteln, während deren ein bestimmtes Interaktionsgerät verwendet werden kann. Dabei wird jeder Trace separat betrachtet. Gemäß der Unterscheidung zwischen passivem und aktivem Ressourcenbedarf im Interaktionsgerätemodell wird auch hier zwischen diesen beiden Zuständen unterschieden. Dabei ist die Kompatibilität des passiven Ressourcenbedarfs mit einem Trace eine notwendige Voraussetzung für die Kompatibilität eines der aktiven Gerätemodi, jedoch nicht umgekehrt. Das bedeutet, dass ein Interaktionsgerät nur dann bedienbar ist, wenn es auch getragen werden kann. Umgekehrt muss ein tragbares Gerät aber nicht bedienbar sein.

Beispiel 23

Ein Touchpad, das am Gürtel getragen wird, kann während einer Endoskopieuntersuchung zwar getragen werden, der Arzt kann es aufgrund der Hygienebestimmungen jedoch nicht bedienen, solange er Handschuhe trägt.

Algorithmus Zunächst wird eine Instanz des Benutzermodells mit dem initialen Ressourcenbedarf des Traces initialisiert. Dann wird das passive Ressourcenprofil des Interaktionsgeräts auf das Modell angewendet und die Konsistenz überprüft. Ist das Benutzermodell bereits zu diesem Zeitpunkt inkonsistent, dann kann das Interaktionsgerät aufgrund der Startvoraussetzungen in diesem Trace nicht getragen werden. Danach wird jedes Ereignis des Traces in ein Start- und ein End-Teilereignis aufgeteilt, wie in Abbildung 4.13 dargestellt ist. Diese Liste wird der Zeit nach aufsteigend sortiert. Das Resultat ist eine Sequenz von Teilereignissen, denen jeweils eine Primäraufgabe und damit ein Ressourcenprofil zugeordnet ist. Diese Ressourcenprofile werden nun der Reihe nach der Simulation hinzugefügt oder aus der Simulation entfernt, wobei nach jedem Schritt die Konsistenz des Benutzermodells überprüft wird. Das weitere Vorgehen wird durch folgenden Pseudocode beschrieben:

```
func berechne_kompatibilität(Gerät G, Trace T):

initialisiere_simulation(S, T)
addiere_ressourcenprofil(G.passiv, S)

If ist_inkonsistent(S):
  return nicht_kompatibel

For E in T.teilereignisse:
  If ist_start(E):
    S.addiere(E.ressourcenprofil)
  else:
    S.substrahiere(E.ressourcenprofil)

  If ist_inkonsistent(S):
    return nicht_kompatibel
```

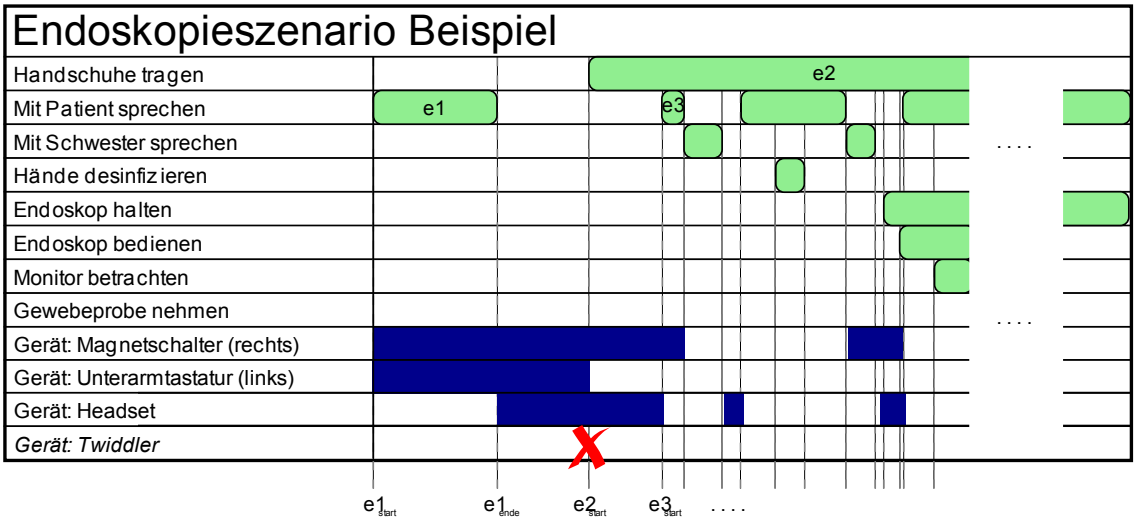

Abbildung 4.13: Die Ereignisse $e_{1..n}$ werden jeweils in ein Start- und ein End-Teilereignis geteilt. Die Liste der Teilereignisse wird sortiert und der Reihe nach abgearbeitet. Nach jedem Schritt wird die Konsistenz des Benutzermodells überprüft. Der Twiddler ist nicht kompatibel, weil das Anziehen der Handschuhe einen Konflikt auslöst.

```
  For F in G.funktionalitäten:
    If S.ist_kompatibel_mit(F.ressourcenprofil):
      Funktionalität F kann bis zum nächsten
            Teilereignis E+1 verwendet werden
  Loop
Loop

return ist_kompatibel
```

Das Ergebnis dieser Analyse ist die binäre Entscheidung für jedes Interaktionsgerät, ob dessen passiver Ressourcenbedarf in diesem Trace gedeckt ist, sowie eine Liste von Zeitbereichen, in denen die einzelnen Funktionalitäten auch benutzbar sind. Diese Zeitbereiche werden als Ereignisse innerhalb des Traces gespeichert, um für weitere Analysen und Visualisierungen zur Verfügung zu stehen.

Ermittlung möglicher Interaktionsstrategien

Ob sich eine Makro-Computeraufgabe mithilfe einer Gerätekonfiguration implementieren lässt, kann ebenfalls automatisch ermittelt werden. Dabei wird auf das gespeicherte Designwissen in Form von Interaktionsstrategien zurückgegriffen, wie Abbildung 4.14 verdeutlicht.

Zunächst wird für alle Interaktionsstrategien, die eine der generischen Computeraufgaben der Makro-Computeraufgabe umsetzten, ermittelt, welche Datenkanäle der aus der gewählten Gerätekonfiguration verfügbaren Interakionsgeräte für eine Implementierung in Frage kommen. Anschließend muss eine Konfiguration von In-

Abbildung 4.14: Bei gegebener Makro-Computeraufgabe und gegebenen Datenkanälen muss eine Implementierung mit Interaktionsstrategien gefunden werden.

teraktionsstrategien gefunden werden, die alle generischen Computeraufgaben implementiert, ohne Datenkanäle doppelt zu verwenden. Dazu kann einfaches Backtracking verwendet werden, da nur so lange gesucht werden muss, bis eine Lösung gefunden ist. Wird eine solche Konfiguration gefunden, ist die Menge der Interaktionsgeräte ausreichend um die Makro-Computeraufgabe zu implementieren. Die verwendeten Interaktionsstrategien wiederum geben dem Designer Aufschluss darüber mit welchen Methoden solch eine Implementierung möglich wäre.

Kann keine Konfiguration gefunden werden, hat der Designer drei Möglichkeiten zu einem positiven Ergebnis zu kommen:

- Die Makro-Computeraufgabe kann verändert werden, indem andere oder weniger generische Computeraufgaben verwendet werden.
- Die Bibliothek der Interaktionsstrategien kann um neue Methoden erweitert werden, die mit anderen oder weniger Datenkanälen auskommen.
- Es werden zusätzliche oder andere Interaktionsgeräte ausgewählt.

4.7.2 Visualisieren

Nach der Analyse tritt die Visualisierung in den Vordergrund. Hier kann sich der Designer ein Bild von der modellierten und analysierten Arbeitssituation machen. Dazu werden in die, bereits für die Modellierung verwendete Zeitleistendarstellung der beobachteten und künstlichen Traces, die Informationen aus der Kompatibilitätsanalyse integriert. Dem Diagramm wird für jedes Interaktionsgerät, das während eines Traces zumindest tragbar ist, eine weitere Zeile für jede seiner Funktionen hinzugefügt. Ereignisse in dieser Zeile zeigen dann an, in welchen Zeitbereichen die Verwendung der jeweiligen Funktionalität möglich wäre, wie Abbildung 4.15 verdeutlicht.

Des Weiteren ist zur besseren Übersichtlichkeit eine aggregierte Darstellung einzelner Abschnitte verschiedener Traces möglich. Abbildung 4.16 zeigt beispielsweise die aggregierte Darstellung auf Basis einer Makro-Computeraufgabe. Hier werden aus allen Traces diejenigen Teilabschnitte hintereinander dargestellt, die dieser Makro-Computeraufgabe zugeordnet sind. Diese Darstellung ermöglicht den Vergleich aller vorkommenden Situationen, in denen diese Makro-Computeraufgabe durchgeführt werden soll. Die Informationen über kompatible Interaktionsgeräte

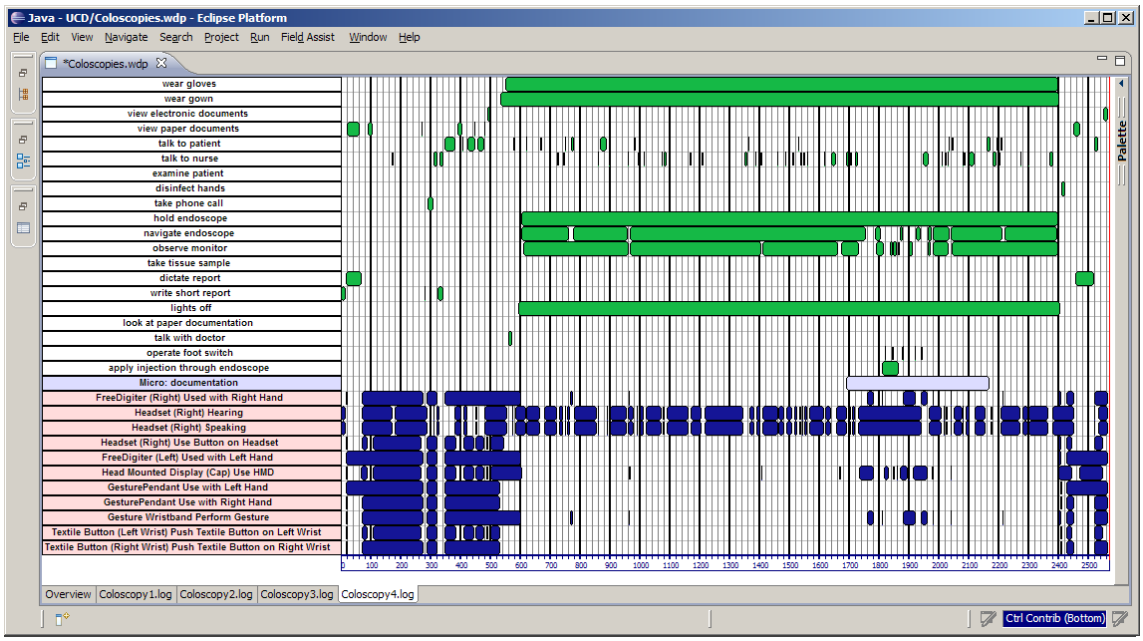

Abbildung 4.15: Visualisierung eines einzelnen Traces im WearableDesigner mit integrierter Kompatibilität der Interaktionsgeräte (dunkelblau). Die Position von Makro-Comptueraufgaben wird ebenfalls dargestellt (hellblau)

werden dabei ebenfalls dargestellt, sodass eine Analyse möglicher Interaktionsgeräte für eine Makro-Computeraufgabe vereinfacht wird.

4.7.3 Modifizieren

Hat der Designer auf Basis der Visualisierung Entwurfsentscheidungen getroffen, müssen diese im Arbeitssituationsmodell umgesetzt werden. Die Änderungen werden dann erneut analysiert und die Auswirkungen können wiederum visualisiert werden. WerableDesigner und ProjectEditor stellen dazu die folgenden Funktionen zur Verfügung:

1. Primäraufgabe modifizieren (hinzufügen/löschen/verändern von Elementen)
2. Traces modifizieren (hinzufügen/löschen/verändern von Ereignissen)
3. Computeraufgabe modifizieren (hinzufügen/löschen/verändern von Makro-/Mikro-Computeraufgaben)
4. Gerätekonfigurationen erstellen und modifizieren

Änderungen an den Eigenschaften der Primäraufgabe lassen sich mit dem ProjectEditor bewerkstelligen. Der WearableDesigner ermöglicht hier nur einige grundlegende Operationen. Änderungen an Traces und den enthaltenen Ereignissen lassen sich hingegen vollständig im WearableDesigner durchführen, wie bereits im Abschnitt 4.6 besprochen. Die Bearbeitung der Computeraufgabe findet ebenfalls überwiegend im ProjectEditor statt. Lediglich die zeitliche Anordnung der Computeraufgabe innerhalb der Traces wird mit dem WearableDesigner durchgeführt. Die Erstellung und

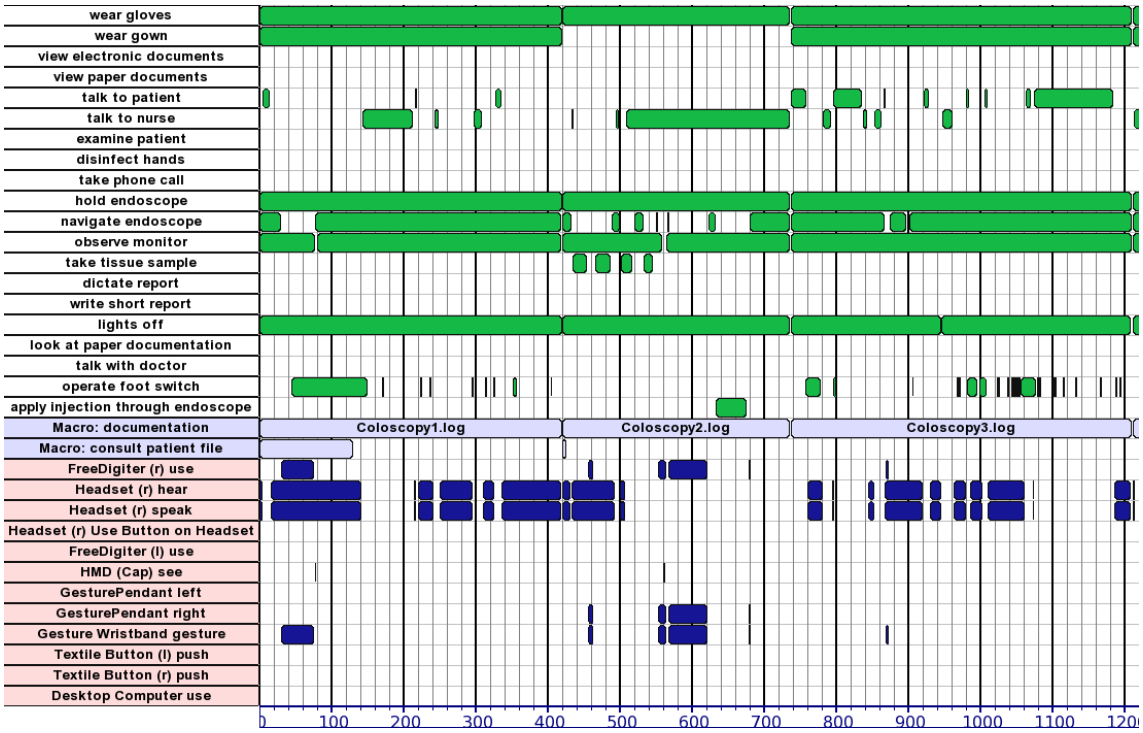

Abbildung 4.16: Aggregierte Ansicht aller Informationen zu einer Makro-Computeraufgabe (hier „documentation“).

Modifikation von Gerätekonfigurationen schließlich wird ebenfalls im ProjectEditor erledigt. Da die Änderungen direkt am zugrunde liegenden Modell durchgeführt werden, ist sofort eine erneute Analyse möglich, sodass die Änderungen direkt wieder visualisiert werden können.

4.8 Zusammenfassung

In den vorangegangenen Abschnitten wurden nun die in Kapitel 3 beschriebenen Konzepte und Modelle vertieft. Das Benutzermodell ermöglicht die Beschreibung des Interaktionsressourcenbedarfs von Primäraufgaben und Interaktionsgeräten und ermöglicht so die Simulation einer Arbeitssituation. Dabei können Wearable-Computing-spezifische Anforderungen an den Entwurf automatisch berücksichtigt werden. Das Interaktionsgerätemodell beschreibt die Gerätefähigkeiten mithilfe eines Kanalmodells, das auf existierenden Modellen aufbaut. Im Gegensatz zu diesen Modellen werden jedoch auch ergonomische Merkmale der Datenkanäle und verschiedenartige Beziehungen zwischen den Datenkanälen berücksichtigt. Hinzu kommen die Benutzeranforderungen, die den aktiven und passiven Ressourcenbedarf eines Interaktionsgerätes beschreiben und die Simulation in einer Arbeitssituation ermöglichen. Diese Simulation erlaubt eine automatische Analyse einer gegebenen Arbeitssituation bezüglich der in Frage kommenden Interaktionsgeräte und erleichtert dem Designer damit den komplexen Auswahlprozess.

Die in dieser Arbeit prototypisch implementierten Werkzeuge TaskObserver,

WearableDesigner und ProjectEditor unterstützen den in Kapitel 3 vorgestellten Entwurfsprozess an verschiedenen Stellen. Der TaskObserver ermöglicht die computergestützte Sequenzanalyse im ersten Prozessschritt. Der WearableDesigner unterstützt die Darstellung und Bearbeitung der so erhobenen Traces und ist in der Lage, die Analyseergebnisse der beschriebenen Algorithmen in diese Darstellung integriert zu visualisieren. Der ProjectEditor schließlich erlaubt die Erstellung und Bearbeitung aller weiteren Modellelemente, insbesondere der Ressourcenprofile.

Das folgende Kapitel wird nun die vorgeschlagenen Modelle, Prozesse und Werkzeuge anhand durchgeführter Wearable-Computing-Projekte evaluieren.

Kapitel 5

Evaluation

Nachfolgend werden die in den vorangegangenen Kapiteln beschriebenen Modelle sowie der dazugehörige Prozess evaluiert. Dazu werden die in Kapitel 2 definierten Teilziele anhand ihrer Anforderungen überprüft. Die Evaluation ist dabei dreigeteilt. Erstens werden die beschriebenen Modelle mit denen der nächstverwandten Arbeit, dem ICE-Tool von Bürgy verglichen. Auf diesem Wege können die Anforderungen theoretisch evaluiert werden. Zweitens wird mit zwei realen Wearable-Computing Projekten der Einfluss der entwickelten Entwurfsunterstützung auf den benutzerorientierten Entwicklungsprozess untersucht. Dies ist die praktische Evaluierung der Anforderungen. Drittens wird eine Benutzerstudie beschrieben, die die computergestützte Sequenzanalyse mit der videogestützten Sequenzanalyse vergleicht und die Benutzbarkeit des hier entwickelten TaskObserver untersucht.

5.1 Vergleich mit dem Modell des ICE-Tools

Die hier vorgestellten Modelle erweitern die Modelle von Bürgy um einige wesentliche Eigenschaften. Im Folgenden werden die Teilmodelle der Arbeitssituation, des Benutzers und des Computersystems mit den äquivalenten Teilmodellen von Bürgy verglichen.

5.1.1 Benutzermodell

Das Benutzermodell von Bürgy ordnet dem Benutzer eine feste Anzahl von Fähigkeiten zu, die jeweils belegt, teilweise belegt oder frei sein können. Die Fähigkeiten sind: Taktile, visuelle und akustische Wahrnehmung; mimische, sprachliche und motorische Fähigkeit sowie Mobilität. Die taktile Wahrnehmung modelliert, ob der Benutzer beide, eine oder keine Hand frei hat. Die anderen Fähigkeiten sind entweder nicht eingeschränkt, teilweise eingeschränkt oder blockiert.

Bürgy's Benutzermodell beschreibt dabei immer nur einen Zustand des Benutzers. Es kann daher, im Gegensatz zu dem Benutzermodell dieser Arbeit, nicht zur Simulation einer Arbeitssituation verwendet werden. Außerdem unterscheidet sich die Granularität der beiden Modelle zum Teil erheblich.

Die *visuelle und die akustische Wahrnehmung sowie die sprachlichen Fähigkeiten* werden bei Bürgy mit 3 Zuständen, anstatt der hier verwendeten 2 Zustände beschrieben. Diese Unterscheidung wird notwendig, da ein großer Zeitbereich durch

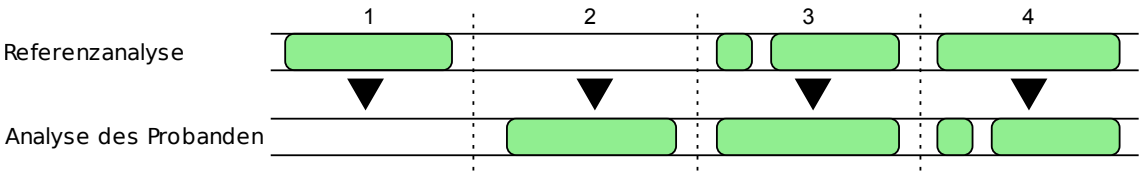

Abbildung 5.1: Vergleich der Modellierung durch Traces mit der Klassifizierung von Bürgy. (1)kein Ereignis bedeutet nicht eingeschränkt (2)einige Ereignisse bedeuten teilw. eingeschränkt (3)ein durchgehendes Ereignis bedeutet blockiert.

einen einzigen Zustand beschrieben wird. Das hier vorgestellte Modell ist durch die Simulation in der Lage, zeitlich feiner aufzulösen und so eine teilweise Verfügbarkeit der drei Eigenschaften direkt zu beschreiben, wie in Abbildung 5.1 dargestellt.

Die *mimische Fähigkeit und die motorische Fähigkeit* beschreiben jeweils einen Teilaspekt der motorischen Fähigkeiten des Benutzers. Dabei wird davon ausgegangen, dass der Benutzer mit den Händen interagiert. Das hier beschriebene Benutzermodell schließt diese beiden Eigenschaften deshalb ein, ist gleichzeitig jedoch in der Lage auch andere Körperteile, wie die Beine als motorische Fähigkeit zu verstehen. Außerdem kann durch die Anforderungen zwischen der Interaktion, alleine durch Bewegung und der Interaktion durch Berührung unterschieden werden.

Die *taktile Wahrnehmung* wird von Bürgy wieder auf den gesamten Körper bezogen. Eine Unterscheidung verschiedener Körperbereiche, wie in dem Modell dieser Arbeit, ist nicht möglich.

Die *Mobilität* modelliert die Bewegungsfreiheit des Benutzers. Diese kann zum Beispiel durch das Tragen eines schweren Gegenstandes beeinträchtigt werden. Die Mobilität wird durch das hier beschriebene Benutzermodell jedoch nicht abgedeckt, da sie für die betrachteten Szenarien und Interaktionsgeräte keine Rolle spielt. Eine Erweiterung des Benutzermodells um den Aspekt der Mobilität ist jedoch prinzipiell möglich.

Das Tragen von Kleidungsstücken und Geräten wird von Bürgy's Benutzermodell nicht abgebildet. Dementsprechend ist es mit dem Modell auch nicht möglich, Aussagen über die Tragbarkeit eines Interaktionsgerätes zu machen. Die hier beschriebene Modellierung kann diese Aspekte hingegen abbilden und erfüllt damit zumindest teilweise Anforderung A6.

Aufgrund der beschriebenen Einschränkungen können Primäraufgaben des Benutzers mit dem ICE-Tool nur sehr grobgranular modelliert werden. Das hier beschriebene Benutzermodell erlaubt hingegen, eine sowohl zeitlich als auch inhaltlich präzisere Modellierung. Aus diesem Grund kann die Primäraufgabe des Benutzers auch effektiver beim Entwurf berücksichtigt werden, als dies mit dem ICE-Tool der Fall ist (A1). Insbesondere die zeitliche Präzision ermöglicht erstmals die Einbeziehung von Multitasking-Aspekten in eine Entwurfsunterstützung, da gleichzeitige Primäraufgaben modelliert werden können (A7). Allerdings ist eine automatische Analyse dieser Aspekte derzeit nicht vorhanden, weshalb diese Anforderung nur teilweise erfüllt wird.

5.1.2 Arbeitssituationsmodell

Das Modell der Arbeitssituation ist bei Bürgy in drei Teile untergliedert: Das *Aufgabenmodell*, das *Umgebungsmodell* und das *Anwendungsmodell*. Das Aufgabenmodell und das Umgebungsmodell entsprechen dem Primäraufgabenmodell, und das Anwendungsmodell entspricht dem Computeraufgabenmodell. Bürgy beschreibt jedoch kein Gegenstück zum Modell des zeitlichen Ablaufs, da Arbeitssituationen bei ihm immer als Ganzes betrachtet werden. Bei der Modellierung muss man sich also entscheiden welchen Zustand man beschreiben möchte: Den günstigsten, den ungünstigsten oder den vorherrschenden Zustand. Das Modell des zeitlichen Ablaufs erlaubt es hingegen, die Variationen innerhalb einer Arbeitssituation detailliert zu beschreiben und zu analysieren.

Primäraufgabe

Bürgy's Aufgabenmodell beschreibt die folgenden Aspekte der Primäraufgabe:

- Benutzer benötigt zusätzliche Werkzeuge (ja/nein)
- Primäraufgabe fordert volle Aufmerksamkeit (ja/nein)
- Benutzer interagiert mit anderen Personen (ja/nein)
- Primäraufgabe benötigt Daten aus einem IT-System (ja/nein)

Die *Verwendung zusätzlicher Werkzeuge* wird von Bürgy pauschal modelliert. Verschiedene Werkzeuge und deren Ressourcenbedarf können im Gegensatz zum hier beschriebenen Modell nicht modelliert werden. Dadurch wird es unmöglich die unterschiedlichen Auswirkungen verschiedener Werkzeuge auf die Interaktionsmöglichkeiten des Benutzers zu modellieren (z. B. Fußschalter und Endoskop im Endoskopieszenario).

Die *Aufmerksamkeit des Benutzers* wird im Gegensatz zu Bürgy's Modell nur implizit durch die Verfügbarkeit der verschiedenen Interaktionsressourcen modelliert. Nimmt eine Primäraufgabe viele Interaktionsressourcen in Anspruch, sodass keine Interaktionsgeräte mehr verwendet werden können, wird auch die Aufmerksamkeit des Benutzers in Anspruch genommen.

Die *Interaktion mit anderen Personen* wird ebenfalls implizit über die Verfügbarkeit der akustischen Wahrnehmung bzw. der sprachlichen Fähigkeit modelliert. Soziale Aspekte von Wearable Computern, die den Umgang mit anderen Menschen beeinflussen können, werden derzeit nicht abgebildet, da diese Aspekte nur schwer zu verallgemeinern sind und stark von dem jeweiligen Szenario abhängen.

Ob die *Primäraufgaben Daten aus einem IT-System benötigt*, wird in dem hier vorgestellten Modell über das *Modell des zeitlichen Ablaufs* beschrieben. Hier werden Primäraufgabe und Computeraufgabe in eine zeitliche Beziehung gebracht und somit ist ersichtlich, welcher Teil der Arbeitssituation Informationen aus einem IT-System benötigt. Im Gegensatz zu Bürgy's Modell können hier jedoch verschiedene Kombinationen von Primär- und Computeraufgaben innerhalb einer Arbeitssituation berücksichtigt werden.

Der Vergleich zeigt, dass die hier entwickelte Modellierung auch andere Aspekte der Arbeitssituation, wie z. B. den Zugriff auf Computersysteme, präziser modellieren kann, als dies mit dem ICE-Tool möglich ist. Dies wiederum erlaubt eine bessere Einbeziehung der Primäraufgabe in den Entwurfsprozess (A1).

Computeraufgabe

Die Computeraufgabe wird von Bürgy durch das Anwendungsmodell beschrieben. Dieses Modell unterscheidet zwischen verschiedenen Medien, die durch das Computersystem ein- oder ausgegeben werden müssen: Text, Tabellen, Diagramme, Skizzen, Fotos, Videos und Audio. Eine Unterscheidung zwischen Ein- und Ausgabe wird dabei ebenso wenig gemacht, wie die Art der Manipulation die möglich sein muss. Die folgenden drei sehr unterschiedlichen Anwendungstypen werden bei Bürgy's Anwendungsmodell beispielsweise identisch modelliert:

- Anzeige einer CAD-Zeichnung aus einer festen Perspektive. Keine Manipulation durch den Benutzer.
- Dynamische Anzeige einer CAD-Zeichnung. Der Benutzer kann die Perspektive bestimmen, das Modell jedoch nicht verändern.
- Bearbeiten einer CAD-Zeichnung. Der Benutzer kann sowohl die Perspektive verändern als auch Änderungen am Modell vornehmen.

Aus diesem Beispiel ist leicht zu ersehen, dass eine statische Modellierung der Anwendungsaufgabe nicht sinnvoll ist. Deshalb sind die möglichen generischen Computeraufgaben des hier beschriebenen Modells nicht vorgegeben und können an das Szenario angepasst werden. Die drei oben genannten Computeraufgaben können z. B. mit einer Kombination aus drei generischen Computeraufgaben modelliert werden: (1)Anzeige eines 3D-Modells, (2)3D-Navigation und (3)3D-Modelle editieren.

5.1.3 Computersystemmodell

Bürgy's Modell des Computersystems besteht lediglich aus einem Gerätemodell, das die Fähigkeiten und die physischen Ausmaße eines Interaktionsgerätes beschreibt. Eine Beschreibung der generischen Computeraufgaben, die mit einem Gerät implementiert werden können, ist nicht vorgesehen. Das Gerätemodell modelliert Interaktionsgeräte durch die folgenden Eigenschaften:

- taktile/visuelle/akustische Ausgabe (ja/nein)
- taktile/visuelle/akustische Eingabe (ja/nein)
- physische Ausmaße (Telefon, PDA, Wearable Computer, Clipboard-Größe, Notebook-Größe)

Das hier beschriebene Interaktionsgerätemodell ist in der Lage, all diese Eigenschaften abzubilden. Die Ein- und Ausgabefähigkeiten werden als Datenkanäle mit entsprechenden Benutzeranforderungen modelliert. Die Kombination aus Gerätefunktionalitäten und Benutzeranforderungen ist allerdings in der Lage viel genauer zwischen verschiedenen Geräten zu differenzieren, als dies mit dem Modell von Bürgy möglich ist, wie folgendes Beispiel verdeutlicht:

Beispiel 24

Zwei verschiedene Geräte, der Magnetschalter und der Twiddler werden in Bürgy's Modell identisch modelliert. Beide Geräte ermöglichen eine taktile Eingabe und haben in etwa die Größe eines Mobiltelefons. In der Praxis gibt es jedoch viele Unterschiede zwischen den beiden Geräten, die durch das hier vorgestellte Modell erfasst werden können. Während der Magnetschalter lediglich die Funktionalität eines einfachen Knopfes zur Verfügung stellt, kann der Twiddler als vollständiger Ersatz für Maus und Tastatur eingesetzt werden. Sein Einsatzgebiet ist also viel größer als das des Magnetschalters. Der Magnetschalter ermöglicht hingegen eine berührungslose Interaktion, die mit dem Twiddler nicht möglich ist. Außerdem wird der Twiddler in der Handfläche getragen und ist daher mit vielen Primäraufgaben nicht kompatibel. Ein Magnetschalter wird am Handgelenk befestigt und kann ohne Konflikte in vielen Situationen verwendet werden.

Wie dieses Beispiel zeigt, ist die Verwendung des Benutzermodells zur Gerätebeschreibung wesentlich ausdrucksstärker und damit flexibler als die einfache Beschreibung von Bürgy. Dadurch wird es möglich, alle in Kapitel 2 beschriebenen Interaktionsgeräte zu modellieren und deren individuelle Eigenschaften zu unterscheiden (A5). Da jedem Interaktionsgerät ein Körperbereich zugeordnet wird, an dem es getragen wird, können durch das vorliegende Modell zusätzlich einige Aspekte der Wearability überprüft werden (A6). Wesentliche Aspekte der Wearability wie bspw. Druck, Einengung und Flüssigkeitstransport werden durch das Modell nicht abgedeckt. In der Praxis kann jedoch meist davon ausgegangen werden, dass diese Aspekte bereits beim Entwurf des Interaktionsgeräts berücksichtigt wurden. Trotzdem wird die Anforderung A6 nur teilweise erfüllt.

5.1.4 Prozess

Es ist zwar möglich eine Arbeitssituation mit Hilfe des ICE-Tools zu modellieren, jedoch können die konkret enthaltenen Abläufe nur schwer, ohne zusätzliche Hilfsmittel, an Dritte kommuniziert werden. Dies liegt zum Einen an der grobgranularen Modellierung vieler Aspekte der Arbeitssituation und zum Anderen an der fehlenden Information über zeitliche Abläufe. In beiden Punkten ist die hier vorgestellte Methodik dem ICE-Tool überlegen (A9).

Des Weiteren erleichtert das ICE-Tool nur dann die Auswahl geeigneter Interaktionsgeräte für eine Arbeitssituation, wenn bereits ein System mit ähnlichen Rahmenbedingungen in der Datenbank gespeichert ist. Die gezielte Auswahl einzelner Interaktionsgeräte wird wiederum durch die unzureichende Ausdrucksstärke der Modellierung erschwert. Wie gezeigt wurde, können einige sehr unterschiedliche Interaktionsgeräte nicht durch diese unterschieden werden, was den Wert eines Vorschlags

deutlich reduziert. In diesem Punkt ist die vorgestellte Methodik dem ICE-Tool also überlegen (A2).

5.1.5 Diskussion

Der Vergleich der beiden Konzepte zeigt deutlich die Schwächen von Bürgy's Modellen, die durch die hier beschriebenen Modelle behoben werden. Das *Benutzermodell* erlaubt die detailliertere Modellierung einiger für Wearable Computing wesentlichen Aspekte des Benutzers, so wie die motorischen Fähigkeiten und die taktile Wahrnehmung. Außerdem wird die Modellierung der Kleidung ermöglicht, was eine einfache Berücksichtigung der Wearability (A6) ermöglicht. Dazu erlaubt das hier beschriebene Benutzermodell Simulationen einer Arbeitssituation, während Bürgy's Benutzermodell statisch ist.

Das *Arbeitssituationsmodell* erfasst ebenfalls Aspekte die von Bürgy's Modellen nicht abgedeckt werden. Die Integration zeitlicher Abläufe ermöglicht beispielsweise, den gezielten Entwurf für bestimmte Situationen innerhalb einer Arbeitssituation. Bürgy hingegen beschreibt eine Arbeitssituation nur mit einem statischen Zustand. Weiterhin erlaubt das Primäraufgabenmodell die detaillierte Modellierung von Arbeitskleidung und Werkzeugen des Benutzers sowie deren Auswirkungen auf seine Interaktionsfähigkeit. Die Computeraufgabe schließlich wird nicht nur durch das Medium der Information beschrieben, sondern auch durch die Art der erforderlichen Interaktion. Zusammen mit der flexiblen Beschreibung aus Makro- und Mikro-Computeraufgaben, lassen sich so die unterschiedlichen Anforderungen realer Aufgaben besser abbilden. Das vorliegende Arbeitssituationsmodell ist also mächtiger als das von Bürgy. Daher kann davon ausgegangen werden, dass die eingangs definierte Anforderung A1 (Berücksichtigung der Arbeitssituation) besser erfüllt wird als durch das ICE-Tool.

Das *Computersystemmodell* schließlich enthält ebenfalls einige wesentliche Verbesserungen gegenüber der Gerätebeschreibung von Bürgy. Im Gegensatz zu dessen Modell, werden auch die Benutzeranforderungen eines Interaktionsgerätes beschrieben. Die Gerätefähigkeiten, die bei Bürgy lediglich durch die verwendeten Modalitäten beschrieben werden, können mit dem Benutzermodell detailliert modelliert werden. Ein Beispiel zeigt, dass mehrere sehr unterschiedliche Interaktionsgeräte, die bei Bürgy identisch modelliert werden, mit dem hier beschriebenen Modell sehr wohl unterschieden werden können und auch unterschieden werden müssen.

Insgesamt sind die in dieser Arbeit entwickelten Modelle besser in der Lage wesentliche Informationen, die für den Entwurf eines Wearable Computing Systems notwendig sind, zu modellieren. Diese verbesserte Dokumentation kann zum einen die Kommunikation innerhalb eines Entwicklungsteams verbessern, zum anderen aber auch für automatische Analysen genutzt werden.

Tabelle 5.1 fasst den Vergleich des ICE-Tools mit der vorliegenden Arbeit zusammen. Die beschriebene Methodik ist dem ICE-Tool in allen Punkten überlegen, wenn man berücksichtigt, dass die Abdeckung der Szenarien nur unter Verwendung der grobgranularen Modellierung des ICE-Tools möglich ist, die einige wichtige Informationen nicht abbilden kann. Um zusätzlich den praktischen Nutzen der beschriebenen Methode nachzuweisen, wird nun der Entwurfsprozess zweier Wearable-Computing-Projekte mit und ohne Verwendung der Methodik verglichen.

Nr.	Name der Anforderung	ICE-Tool	vorl. Arbeit
A 1	Berücksichtigung der Arbeitssituation	□	■
A 2	Auswahl geeigneter Interaktionsgeräte	□	■
A 5	Abdeckung verschiedener Interaktionsgeräte	□	■
A 6	Berücksichtigung von Wearability		□
A 7	Berücksichtigung der Auswirkungen von Multitasking		□
A 9	Kommunikation der Arbeitssituation	□	■

Tabelle 5.1: Abdeckung der Anforderungen durch die vorliegende Arbeit im Vergleich mit den ICE-Tool. □ = teilweise/unzureichend erfüllt, ■ = erfüllt. (*)A 1 wird durch das ICE-Tool nur im Rahmen der eigenen Modellierung erfüllt.

5.2 Erfahrungen in wearIT@work

Die in dieser Arbeit beschriebene Methode wurde wesentlich durch das wearIT@work Projekt geprägt, in dessen Rahmen sie entwickelt wurde. In diesem Projekt wurden verschiedene Szenarien untersucht, deren Entwicklung zeitversetzt gestartet wurde. Im Bereich Gesundheitswesen war dies zunächst das in Kapitel 2.3 beschriebene Visitenszenario, das mit herkömmlichen Methoden bearbeitet wurde. Später wurde parallel das Endoskopieszenario, mit Hilfe erster Prototypen der hier verwendeten Methode untersucht.

Beide Szenarien wurden unter Verwendung eines ISO 13407 kompatiblen benutzerorientierten Entwicklungsprozess bearbeitet. Im Endoskopieszenario wurden zusätzlich Prototypen und Mock-ups der hier vorgestellten Werkzeuge verwendet, um den Entwurf zu unterstützen. Nachfolgend wird zunächst der Entwurfsprozess des Visitenszenarios beschrieben, um anschließend die Auswirkungen der vorgestellten Methode am Beispiel des Endoskopieszenarios zu untersuchen.

5.2.1 Szenario: Visite

Zu Projektbeginn musste zunächst ein geeignetes Szenario für die Unterstützung durch einen Wearable Computer identifiziert werden. Zu diesem Zweck wurde eine eintägige Begehung des Krankenhauses unter Begleitung eines Oberarztes organisiert. Während dieser Begehung wurden die verschiedenen Arbeitsbereiche und Räumlichkeiten des Krankenhausalltags begutachtet. Dort ablaufende Prozesse konnten nur rudimentär beschrieben werden, da eine Beobachtung des tatsächlichen Prozesses mit Patienten nicht möglich war. Aufgrund dieser Begehung wurde die Visite als erstes Szenario ausgewählt.

Anschließend begann der benutzerorientierte Entwicklungsprozess. Um den Nutzungskontext zu erheben, wurden zunächst Beobachtungen der Visite mit vier verschiedenen Ärzten durchgeführt. Da die Visite nur einmal am Tag unter enormem

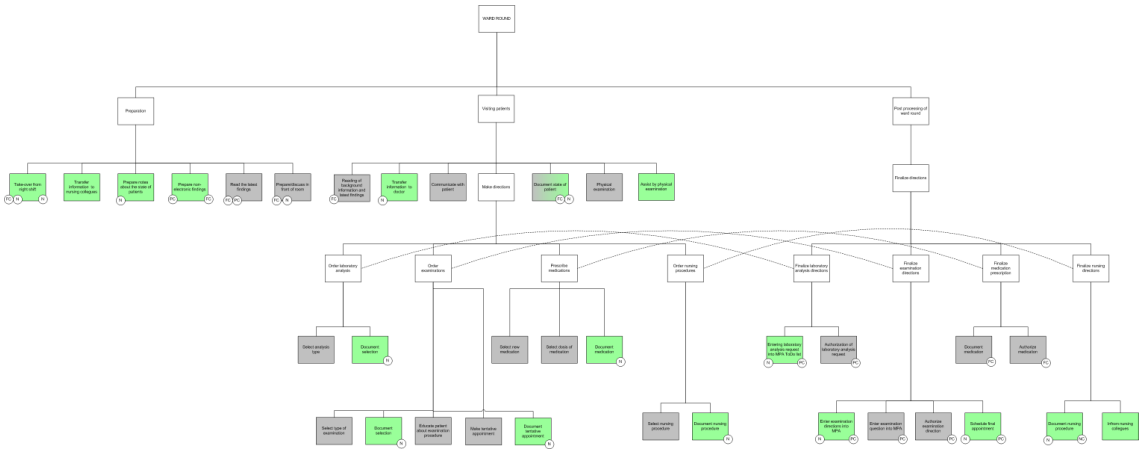

Abbildung 5.2: wearIT@work Visite HTA Modell.

Zeitdruck stattfindet, war eine Unterbrechung der Ärzte um Fragen zu stellen nicht möglich. Außerdem musste die Anwesenheit externer Personen vor den Patienten verheimlicht werden. Deshalb mussten sich die Beobachter als Medizinstudenten ausgeben. Aus diesem Grund war es aber nicht möglich Videoaufzeichnungen durchzuführen, weshalb lediglich Notizen mit Papier und Stift gemacht werden konnten. Diese Notizen wurden anschließend in einem Interview mit dem jeweiligen Arzt besprochen, um Unklarheiten zu beseitigen. Aus diesen Informationen wurde ein HTA-Modell des Visitenprozesses erstellt (siehe Abb. 5.2) und erste Verbesserungspotenziale identifiziert.

Im nächsten Schritt sollte das Modell des Visitenprozesses in Zusammenarbeit mit den Ärzten validiert werden. Dazu wurde ein Arbeitstreffen mit allen beteiligten Ärzten organisiert. Zunächst wurde die HTA-Notation vorgestellt und anschließend das Modell aus Abbildung 5.2 besprochen. Die Ärzte hatten jedoch Schwierigkeiten mit der abstrakten Repräsentation der Visite zu arbeiten. Stattdessen beschrieben sie ihre eigene Sicht der Visite und die aus ihrer Sicht notwendigen Verbesserungen. Aufgrund unterschiedlicher Meinungen kam es dabei schnell zu Diskussionen. Eine Validierung der durch das Entwurfsteam identifizierten Verbesserungspotenziale, auf Basis des Prozessmodells, war so kaum möglich. Dadurch, dass keinerlei verlässliche Daten über die zeitlichen Abläufe der Visite vorlagen, wurde eine Diskussion über die Verbesserungspotenziale zusätzlich erschwert. Die Richtigkeit der beobachteten Situationen wurde durch die Ärzte sogar teilweise angezweifelt. Dies zeigt erneut, wie wichtig es ist eine einfache Repräsentation der Arbeitssituation zu haben, damit alle am Prozess Beteiligten auf derselben Grundlage diskutieren können (A9).

Dasselbe Arbeitstreffen wurde außerdem für die Erhebung organisatorischer und rechtlicher Anforderungen genutzt. Im Gespräch mit den Ärzten und einem hinzugezogenen Juristen des Krankenhauses, wurden Entwurfsideen und Verbesserungspotenziale auf ihre Verträglichkeit mit den rechtlichen Rahmenbedingungen hin überprüft.

Auf Basis der in diesem Arbeitstreffen gewonnenen Erkenntnisse sowie der HTA-Modelle wurde anschließend ein Entwurf generiert und in Form eines Mock-ups implementiert. An diesem Arbeitsschritt war nun eine deutlich größere Personen-

gruppe beteiligt, als an den Besuchen vor Ort im Krankenhaus. Die zusätzlichen Personen mussten sich auf die Modelle und Beschreibungen derjenigen verlassen, die den Visitenprozess aus erster Hand erlebt hatten. Reichte die Dokumentation nicht aus um eine konkrete Frage zu beantworten, mussten die Personen, die direkt an den Beobachtungen beteiligt waren, als Ersatz für einen Benutzer/Arzt fungieren und Fragen zur Arbeitssituation beantworten. Der Mock-up wurde dann mit einigen Ärzten und Pflegern im Krankenhaus getestet. Obwohl diese Tests unter Laborbedingungen durchgeführt wurden, konnten doch einige Probleme dieser ersten Entwurfsideen identifiziert werden. Beispielsweise konnten Inkompatibilitäten zwischen Interaktionsgeräten und medizinischen Werkzeugen festgestellt werden:

Beispiel 25

Teil des Mock-ups war ein Bluetooth Headset. Dieses sollte zur Aufzeichnung von Anforderungen und für Audiorückmeldungen verwendet werden. Die Verwendung des Headsets kollidierte jedoch mit der Verwendung eines Stethoskops, wie es von Internisten zum Abhören eines Patienten verwendet wird. Beide Geräte werden im Ohr platziert und sind deshalb nicht gleichzeitig verwendbar.

Dieser Zusammenhang ist zwar trivial, wurde jedoch trotz existierender Dokumentation von keinem Mitglied des Entwurfsteams erkannt, da das Stethoskop nur in einzelnen Fällen und für kurze Zeit verwendet wurde. Wäre dieser Zusammenhang früher erkannt worden, hätte man bereits für den ersten Prototypen andere Geräte verwendet und so die Anzahl notwendiger Entwurfsiterationen reduzieren können. Dieses Beispiel unterstreicht die Notwendigkeit, die Auswahl der Interaktionsgeräte, wie in Kapitel 2 beschrieben, zu unterstützen (A 2).

Die nächste Iteration begann mit der Auswahl geeigneter alternativer Interaktionsgeräte. Dazu wurden einem Oberarzt verschiedene Interaktionsgeräte vorgestellt. Im Gespräch wurde dann festgestellt, ob und wie gut die jeweilige Technik während der Visite eingesetzt werden könnte. Die Ergebnisse wurden verwendet um einen neuen Entwurf zu entwickeln, der dann als Prototyp implementiert und evaluiert wurde.

Probleme

Die Effektivität des verwendeten Entwurfsprozesses wurde vor allem durch die folgenden Probleme beeinträchtigt:

- HTA-Modelle unterstützen die Diskussion mit Benutzern nur schlecht (A9).
- Zeitliche Aspekte wie Dauer und Zeitpunkt verschiedener Aufgaben werden von HTA-Modellen nicht erfasst (A1,A9).
- Die Dokumentation vermittelte anderen Teammitgliedern lediglich einen groben Überblick des Szenarios (A9).
- Selbst triviale Inkompatibilitäten eines Entwurfs wurden nicht frühzeitig erkannt und führten zu zusätzlichen Entwurfsiterationen (A2).

5.2.2 Szenario: Endoskopie

Nachdem der erste Prototyp des Visitenszenarios getestet wurde, begannen die Arbeiten an einem weiteren Szenario aus dem Krankenhausumfeld. Ausgewählt wurde die ambulante Endoskopieuntersuchung, da der Umgang mit dem Endoskop die Bedienung eines herkömmlichen PC verhindert. Das Szenario wurde bereits in Abschnitt 1.1 ausführlich beschrieben und im Laufe der Arbeit häufig als Beispiel verwendet. Im Gegensatz zum Entwurfsprozess des Visitenszenarios, konnten hier bereits, die in dieser Arbeit entwickelten Verfahren und Werkzeuge, eingesetzt werden, um die während der Entwicklung des Visitenszenarios aufgetretenen Probleme zu vermeiden. Dabei wurden Prototypen der hier vorgestellten Werkzeuge verwendet, um zunächst die Eignung der Visualisierung sowie die Nützlichkeit der Analysen und Werkzeuge zu evaluieren.

Phase 1: Datenerhebung

Der Entwurfsprozess begann mit der Auswahl eines geeigneten Szenarios aus dem Krankenhausumfeld. Zu diesem Zweck wurden Beobachtungstermine mit Ärzten aus drei verschiedenen Abteilungen vereinbart: Unfallabteilung, innere Medizin und ambulante Chirurgie. Die Umstände der Arbeitsumgebungen machten verschiedene Vorgehensweisen notwendig. Die Umgebungen wurden auf die Möglichkeit hin untersucht Videoaufzeichnungen durchzuführen sowie den Benutzer bei der Arbeit zu unterbrechen. Anschließend wurde die jeweilige Vorgehensweise aus Abbildung 3.9 abgelesen:

- **Unfallabteilung**: unterbrechbar, kein Video → Befragung im Kontext
- **innere Medizin**: unterbrechbar, kein Video → Befragung im Kontext
- **ambulante Chirurgie**: nicht unterbrechbar, kein Video → einfache Beobachtung und Interview

Ausgewählt wurde schließlich das Endoskopieszenario aus der ambulanten Chirurgie, das nun im Folgenden weiter behandelt wird. Auf die initialen Beobachtungen folgte ein weiterer Besuch im Krankenhaus zur eingehenderen Untersuchung des Endoskopieszenarios. Ergebnisse der initialen Beobachtungen und Interviews, waren die Grundzüge der Aufgabenstruktur, der Umgebungseinflüsse sowie der Interaktionsressourcen. Diese galt es nun zu vertiefen und zusätzlich Informationen über die zeitliche Abfolge des Szenarios zu sammeln.

Wie auch beim Visitenszenario war es weder möglich Videoaufzeichnungen durchzuführen, noch den Arzt bei der Arbeit zu unterbrechen. Mithilfe des TaskObserver konnten dennoch Traces aufgezeichnet werden. Eine initiale Ereignisliste für die Verwendung des TaskObserver wurde auf der Grundlage der initialen Beobachtung und unter Berücksichtigung von Abschnitt 4.5.1 erstellt. Insgesamt wurden 6 Beobachtungen mit dem TaskObserver Werkzeug aufgezeichnet. Dabei mussten nur wenige neue Ereignisse im Laufe der Beobachtungen hinzugefügt werden, was dafür spricht, dass die verwendeten Auswahlkriterien gut gewählt wurden. Die Erfahrungen im Umgang mit dem Werkzeug wurden außerdem zur Verbesserung der Benutzungsoberfläche eingesetzt. Die bereits vorhandenen Informationen wurden durch die durchgeführten Beobachtungen sowie zusätzliche Interviews erweitert.

Phase 2: Modellierung

Anschließend folgte die Phase der Modellierung. Hier wurden zunächst die gesammelten Traces konsolidiert. Neue Ereignisse wurden in das allgemeine Modell übernommen und zwischen verschiedenen Beobachtungen abgeglichen. Besonderheiten und Notizen zu den Beobachtungen wurden direkt in den Abläufen vermerkt. Abbildung 5.3 zeigt alle 6 konsolidierten Beispiele inklusive Notizen und farblicher Markierung verschiedener Ereignisarten, in ihrer prototypischen Darstellung. Durch die Interviews wurden außerdem zusätzliche Situationen und Geräte identifiziert, die nur selten vorkommen und deshalb nicht beobachtet werden konnten. Künstliche Beispiele mussten dennoch nicht erstellt werden, da sich die Bedienung dieser Geräte (z. B. Argon Laser) in Bezug auf das verwendete Benutzermodell nicht von der des Endoskops unterscheidet.

Anschließend folgte eine erste Analyse der erhobenen Traces, um Verbesserungspotenziale zu identifizieren. Die verwendete Darstellung erleichterte das Erfassen der zeitlichen Zusammenhänge und somit die Analyse. Dies ist ein Indikator dafür, dass die eingesetzte Methode die Anforderung A 1 erfüllt. Das größte Verbesserungspotenzial wurde beim Umgang mit der Dokumentation gesehen. Die Traces zeigten deutlich, dass die Ärzte viele Informationen über einen langen Zeitraum im Gedächtnis behalten müssen, da sie während der Untersuchung nicht auf den PC zugreifen können. Die untersuchungsbegleitende Einsicht der Dokumentation wurde im Folgenden als Entwurfsziel angenommen und gemeinsam mit den Benutzern validiert.

Zu diesem Zweck wurde analog zum Visitenszenario ein Arbeitstreffen mit insgesamt 5 an der ambulanten Endoskopie beteiligten Ärzten durchgeführt. Im Gegensatz zum vorangegangenen Arbeitstreffen waren die Ärzte jedoch besser in der Lage, die Darstellung der Arbeitssituation zu verstehen. Außerdem schufen die konkreten Beispiele Fakten, welche die zielgerichtete Diskussion mit den Ärzten erleichterte. Die Vorstellung und Diskussion des identifizierten Entwurfsziels blieb daher weitestgehend sachlich. Somit wurde das vorgestellte Entwurfsziel bestätigt und mögliche Entwurfsideen konnten besprochen werden. Dies ist ein deutlicher Indikator dafür, dass die gewählte Visualisierung die Kommunikation mit Dritten unterstützt (A9).

Nach der Festlegung des Entwurfsziels wurde dann mit der Modellierung von Primär- und Computeraufgabe begonnen. Zwei Computeraufgaben wurden definiert: *„Dokumentation einsehen“*, bestehend aus der generischen Computeraufgabe *„Navigation“* sowie *„Dokumentation erstellen“*, bestehend aus der generischen Computeraufgabe *„Sprachaufzeichnung“*. Dementsprechend wurden Ereignisse aus der Primäraufgabe entfernt, die zur Erfüllung dieser Aufgaben dienten (z. B. *„Untersuchungsbericht diktieren“*). Anschließend wurden die Ressourcenprofile der übrigen Primäraufgaben, auf Basis der Beobachtungen und Interviews, modelliert.

Phase 3: Entwurf

Mit Abschluss der Modellierung folgte die Entwurfsphase. Die Entwurfsaktivitäten erstreckten sich jedoch auf einen größeren Personenkreis, als bei den ersten beiden Phasen. Deshalb war es notwendig, das Szenario an die neuen Teammitglieder zu kommunizieren. Hier halfen vor allem die Visualisierung des zeitlichen Ablaufs sowie Fotos des Endoskopieraums.

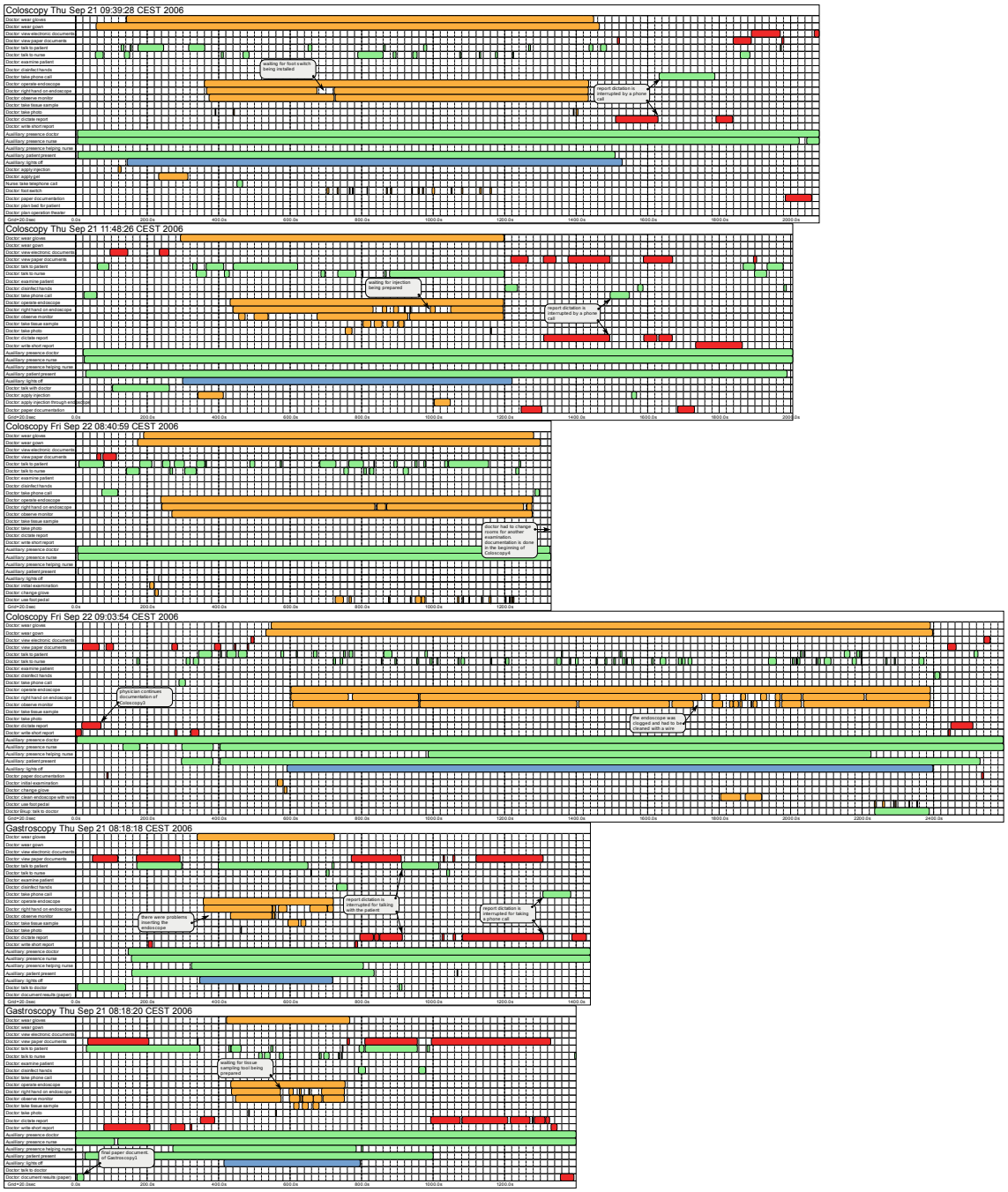

Abbildung 5.3: Konsolidierte Traces des Endoskopieszenarios. Farben unterscheiden die verschiedenen Arten von Ereignissen: Umgebungseinflüsse (blau), Teilaufgaben der Untersuchung (orange), zukünftige Computeraufgaben (rot) und sonstige (grün). Notizen sind direkt in die Darstellung integriert.

Anschließend folgte die Auswahl geeigneter Interaktionsgeräte zur Implementierung der beiden Makro-Computeraufgaben. Hierzu wurde vor allem die automatische Analyse der kompatiblen Interaktionsgeräte genutzt. Dabei wurden Modelle aller in Abschnitt 2.2 beschriebenen Interaktionsgeräte verwendet. Dass es möglich war alle Interaktionsgeräte zu modellieren zeigt, dass die Modellierung die Anforderung A5 erfüllt. Die Visualisierung der Ergebnisse der anschließenden Analyse zeigte deutlich, dass nur wenige Geräte für eine Interaktion während der Untersuchung in Frage kommen. Das Tragen des Kittels und der Handschuhe verhindert die Bedienung der meisten Geräte. Ohne die automatische Analyse scheint zunächst gar kein Interaktionsgerät in Frage zu kommen das die Hände benutzt, da der Arzt das Endoskop mit beiden Händen bedienen muss. Die automatische Analyse berücksichtigt jedoch nicht nur die vorherrschende Situation, sondern alle Zeitpunkte innerhalb eines Ablaufs, was folgende Beobachtung zulässt:

Beispiel 26

Eigentlich hat der Arzt während der Untersuchung keine Hand frei, um mit Geräten zu interagieren. Während der Endoskopieuntersuchung kommt es jedoch vor, dass Gewebeproben entnommen werden. Die beobachteten Beispiele zeigen, dass der Arzt während und kurz nach einer Probenentnahme, die rechte Hand vom Endoskop entfernt. Dies führt dazu, dass Geräte verwendet werden können, die berührungslos mit der rechten Hand bedient werden. So in etwa einen Magnetschalter oder ein Gestenarmband.

Eine Visualisierung wie in Abbildung 3.13 zeigte für das Endoskopieszenario deutlich, dass es Zeitpunkte gibt, zu denen die Interaktion mit dem Gestenarmband oder einem Magnetschalter, zusätzlich zu dem häufig kompatiblen Headset möglich ist. Die zusätzlichen Informationen lassen außerdem darauf schließen, dass der Arzt in der Lage ist, die rechte Hand auch in anderen Situationen kurzfristig vom Endoskop zu entfernen, um mit dem Computer zu interagieren. Aus diesem Grund können diese Interaktionsgeräte für die vorgegebenen Computeraufgaben in Betracht gezogen werden. Eine weitere Information, die sich aus der automatischen Analyse ablesen lässt, ist, dass das GesturePendant während einer Coloskopie nicht verwendet werden kann. Es kann jedoch sehr wohl während einer Gastroskopie verwendet werden, da der Arzt hier keinen Kittel trägt, der die Kamera des GesturePendant verdeckt.

Die automatische Analyse trägt also zum Einen dazu bei, nicht-triviale Interaktionsgeräte zu identifizieren und zum Anderen erledigt sie die Aufgabe alle trivialen Randbedingungen automatisch zu überprüfen, sodass hier keine Anforderungen übersehen werden können. Auf diese Weise werden Probleme mit Entwürfen, wie z. B. die Inkompatibilität von Headset und Stethoskop, ohne zusätzlichen Aufwand identifiziert. Die Beispiele belegen, dass die automatische Komaptibilitätsprüfung den Auswahlprozess geeigneter Interaktionsgeräte unterstützt, womit Anforderung A2 erfüllt wird.

5.2.3 Diskussion

Die Erfahrungen aus dem Endoskopieszenario haben gezeigt, dass der hier entwickelte Entwurfsprozess geeignet ist, die Bewältigung Wearable Computing spezifischer Probleme zu unterstützen. Während der *Datenerhebung* hilft die Prozessbeschreibung dabei, geeignete Methoden der Feldforschung auszuwählen. Insbesondere die computergestützte Sequenzanalyse wird durch den TaskObserver unterstützt. Dasselbe gilt für die *Modellierung*. Die Erfahrungen haben gezeigt, dass die entwickelten Modelle in der Lage sind, alle wesentlichen Aspekte der Arbeitssituation zu beschreiben und zu dokumentieren. Die detaillierte Dokumentation hilft zudem bei der Kommunikation mit den Benutzern und erhöht so die Effektivität benutzerorientierter Methoden. In der *Entwurfsphase* schließlich zeigt sich der Nutzen der automatischen Analyse der Gerätekompatibilitäten. Sie hilft dabei scheinbar triviale Fehler zu vermeiden, indem eine Vielzahl für sich genommener Randbedingungen automatisch überprüft werden. Sie versetzt den Designer auch in die Lage nicht-triviale Interaktionsgeräte, wie z. B. Magnetschalter oder Gestenarmband, zu berücksichtigen. In beiden Fällen kann der zeitliche Ablauf einer Arbeitssituation mit berücksichtigt werden, sodass gezielt für bestimmte Abschnitte einer Arbeitssituation entworfen werden kann.

5.3 Benutzerstudie: Computergestützte Sequenzanalyse

Um die Vor- und Nachteile computergestützter Sequenzanalyse im Vergleich zu einer videogestützten Sequenzanalyse besser zu verstehen, wurde eine Benutzerstudie durchgeführt. Die Studie hatte zwei Hauptziele:

1. Den Fehler einer computergestützten Sequenzanalyse mit dem TaskObserver gegenüber einer videogestützten Sequenzanalyse abschätzen zu können und diesen der gesparten Zeit gegenüberzustellen.

2. Die Fehler, die bei einer computergestützten Sequenzanalyse gemacht werden zu klassifizieren, und die Effekte verschiedener Typen von beobachteten Ereignissen auf die Genauigkeit zu untersuchen.

Diese Ergebnisse sollten es möglich machen besser zu entscheiden, unter welchen Umständen der Mehraufwand einer videogestützten Sequenzanalyse sinnvoll ist und unter welchen Bedingungen die Genauigkeit einer computergestützten Sequenzanalyse ausreicht. Gleichzeitig wurde die Benutzbarkeit der TaskObserver Software evaluiert und anhand der Ergebnisse weiter verbessert.

Aufgabe der Probanden war die Durchführung einer Sequenzanalyse eines vorgegebenen Szenarios. Die Probanden wurden in 2 Gruppen geteilt. Eine der Gruppen benutzte den TaskObserver um eine computergestützte Sequenzanalyse durchzuführen, während die andere eine videogestützte Sequenzanalyse desselben Szenarios durchführte.

Die Benutzerstudie wurde mit zwei Probandengruppen durchgeführt. Die erste Gruppe wurde gebeten, Videos einer Kochsituation mit dem TaskObserver zu

codieren. Die zweite Gruppe verwendete zur Codierung derselben Situationen das Video-Annotationswerkzeug ANVIL [Kip04]. Die Ergebnisse der ersten Teilstudie wurden verwendet, um die Genauigkeit der Sequenzanalyse zu quantifizieren sowie die Benutzbarkeit der TaskObserver Software zu evaluieren. Die zweite Teilstudie diente zur Ermittlung der durchschnittlich benötigten Zeit für die exakte Codierung eines Videos. Damit wird die Abschätzung des Mehraufwandes einer solchen Analyse im Vergleich zu einer computergestützten Sequenzanalyse möglich.

Der Rest dieses Abschnitts ist wie folgt aufgebaut. Zunächst wird die Situation vorgestellt, die in beiden Teilstudien analysiert wurde. Anschließend werden die beiden Versuchsaufbauten genauer beschrieben. Danach werden die Versuchsergebnisse bezüglich Genauigkeit, Mehraufwand und Benutzbarkeit untersucht. Abschließend werden Schlüsse für die Erhebung von Beobachtungsdaten gezogen.

5.3.1 Szenario und Versuchsaufbau

Beide Teilstudien verwenden dasselbe Szenario als Grundlage für die Sequenzanalyse. Die zu beobachtende Szenen zeigen jeweils eine Person bei der Zubereitung eines Wok-Gerichts in einer Küche (siehe Abb. 5.4). Die Struktur jeder einzelnen Szene folgt dabei einem Muster:

> *Zunächst liest der Koch ein Rezept aus einem Kochbuch auf der Arbeitsplatte. Dann setzt er Wasser in einem Topf zum Kochen auf. Anschließend werden verschiedene Zutaten in eine Pfanne auf dem Kochfeld gegeben. Sobald das Wasser kocht, werden Nudeln hinzugefügt. Sind diese fertig, werden sie unter kaltem Wasser abgeschreckt und ebenfalls in die Pfanne zu den restlichen Zutaten gegeben. Ist das Essen schließlich fertig, wir der Inhalt der Pfanne in eine Schüssel gegeben. Die Pfanne und der Topf werden gespült und getrocknet. Parallel zu diesen Basisaufgaben wurden kleinere Aufgaben wie das Einstellen des Herdes oder Einschalten der Dunstabzugshaube ausgeführt.*

Die Situation und die einzelnen Szenen wurden sorgfältig im Hinblick auf die Ziele der Benutzerstudie ausgewählt. Vier dieser Szenen wurden inszeniert und auf Video festgehalten, um dasselbe Erlebnis für alle Teilnehmer zu gewährleisten. Jedes einzelne Szenario unterschied sich von den anderen in der Auswahl der verwendeten Zutaten und auch in der Reihenfolge bestimmter Ereignisse. Jede Szene enthielt zusätzlich Aufgaben, die in keiner anderen Szene vorkamen. Diese zusätzlichen Aufgaben waren etwa das Schneiden von Gemüse oder die Bedienung eines Mobiltelefons. Diese Unterschiede wurden eingebaut um die Varianz und Unvorhersehbarkeit einer realen Beobachtung widerzuspiegeln, wo es selten zu zwei identischen Arbeitsabläufen kommt. Besonders zu Beginn einer Beobachtungsreihe treten häufig bis dahin unbekannte Ereignisse auf.

Die gewählten Szenen besitzen außerdem die Bandbreite verschiedener Ereignisse, die notwendig sind, um die Ziele dieser Studie zu erreichen. Sie enthalten sehr hektische Passagen, in denen mehrere Dinge in schneller Abfolge und auch gleichzeitig geschehen, aber auch ruhige Abschnitte in denen nur wenige Ereignisse auftreten. Von den Teilnehmern wurde verlangt, sowohl sehr kurze Ereignisse (Länge < 4 Sekunden, z. B. Herd einstellen) als auch mittlere und sehr lange Ereignisse (Länge $>$

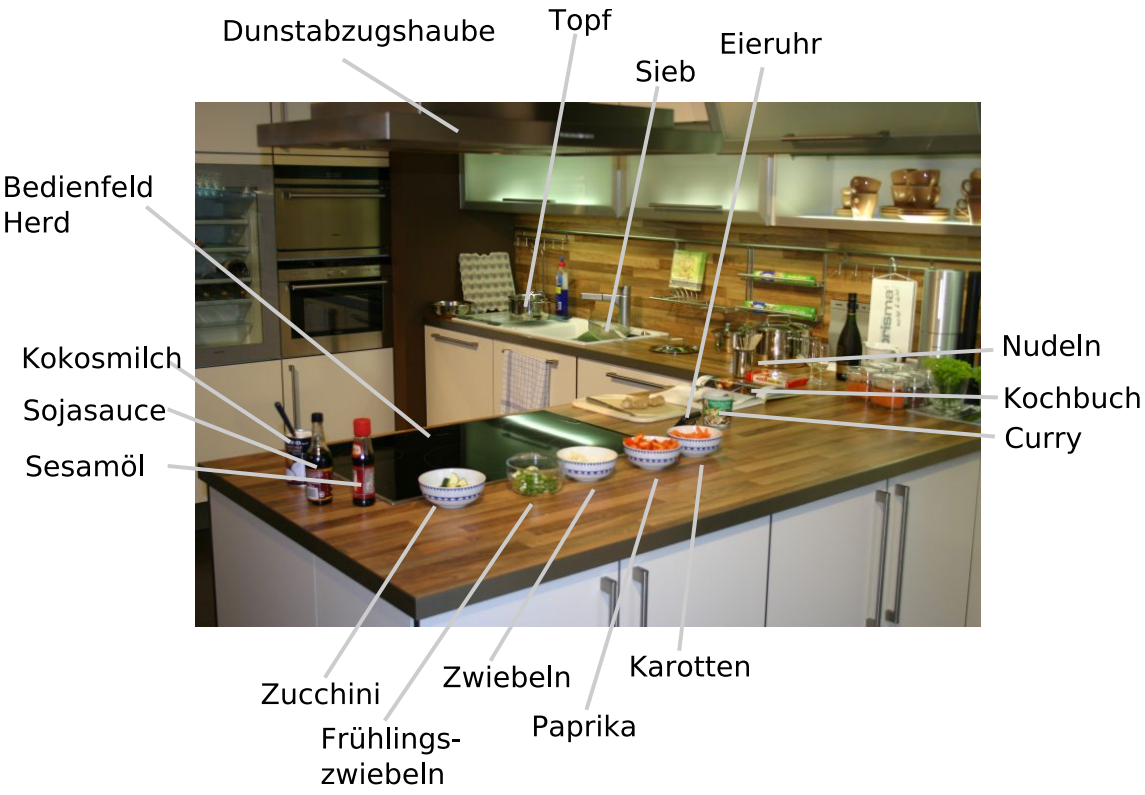

Abbildung 5.4: Übersicht des Kochszenarios, die für die Einführung der Teilnehmer verwendet wurde.

60 Sekunden, z. B. Topf steht auf dem Herd) zu erfassen. Dies ermöglichte herauszufinden, ob die Länge der beobachteten Ereignisse einen Einfluss auf die Genauigkeit hat.

Die Ereignisse variierten zudem in ihrer Offensichtlichkeit für den Beobachter. Sie enthielten primäre Aufgaben des Kochs, sekundäre Aufgaben und Ereignisse, die nur indirekt mit den Aktivitäten des Kochs in Bezug standen. Den Inhalt der Pfanne wenden ist zum Beispiel eine primäre Aufgabe. Das Einschalten der Dunstabzugshaube ist hingegen eine sekundäre Aufgabe, die häufig durchgeführt wird, ohne dass der Darsteller hinsieht. Umgebungsbedingungen wie die Position der Pfanne auf dem Herd repräsentierten das andere Ende des Spektrums an Ereignissen, die nur indirekt mit den Aktivitäten des Kochs in Verbindung stehen.

Außerdem zwangen die jeweils für ein Szenario einzigartigen Ereignisse die Teilnehmer neue Ereignisse in die Ereignisliste aufzunehmen. Dies ermöglichte die Evaluierung dieser speziellen Funktion des TaskObserver und sorgte für ein realitätsnahes Beobachtungserlebnis.

Computergestützte Sequenzanalyse (Teil 1)

Im ersten Teil der Studie wurden die Teilnehmer gebeten die TaskObserver Software zu verwenden, um die vier verschiedenen Kochszenen in Echtzeit zu erfassen. Die Videos wurden in Lebensgröße projiziert, um eine möglichst realistische Beobachtungssituation zu schaffen. Die Verwendung des TaskObservers erfordert eine initiale

Ereignisliste, die für gewöhnlich vor dem Start der eigentlichen Beobachtungen erstellt wird, siehe Abschnitt 4.5.1. Um die Vergleichbarkeit der Ergebnisse zwischen den Teilnehmern und den beiden Teilstudien zu gewährleisten, wurde eine solche Ereignisliste vorgegeben. In der Folge waren die Teilnehmer weder mit der Liste, noch mit dem Kochszenario vertraut. Dies steht im Gegensatz zu einer gewöhnlichen Benutzung der TaskObserver Anwendung, bei der der Benutzer das Szenario für gewöhnlich kennt, da er die initiale Ereignisliste selbst erstellt hat.

Um den Effekt der unbekannten Ereignisliste zu minimieren, wurde die Liste so angelegt, dass sie für unbedarfte Benutzer leicht erlernbar war. Diese Erlernbarkeit wurde vor Beginn der Studie mit weiteren Probanden in mehreren Iterationen erreicht. Das Ergebnis dieser Vorbereitung war eine Liste, bei der Knöpfe mit ähnlichem Inhalt auch räumlich nebeneinander angeordnet wurden (z. B. die Handhabung verschiedener Küchengeräte). Wenn möglich wurden die Knöpfe zusätzlich gemäß ihrer räumlichen oder zeitlichen Anordnung innerhalb der Szenarien angeordnet. Die Knöpfe für das Hinzufügen der diversen Zutaten wurden z. B. analog zu ihrer Position auf der Arbeitsplatte angelegt (siehe Abb. 5.4, Zutaten: Zucchini, Frühlingszwiebeln, Zwiebeln, Paprika und Karotten).

Die Studie selbst wurde folgendermaßen durchgeführt. Zunächst erhielt jeder Proband eine kurze Einleitung in die TaskObserver Software und anschließend in das Kochszenario. Hierzu wurde die Abbildung 5.4 verwendet. Nach dieser Einweisung mussten die Probanden das erste Video erfassen. Dieser erste Durchgang diente der Einarbeitung in die Bedienung der Software und zur Klärung inhaltlicher Fragen bezüglich des Szenarios. Danach wurden die verbleibenden 3 Videos ohne jegliche Hilfestellung erfasst. Abschließend wurden die Probanden zur Bedienbarkeit des TaskObserver befragt.

Bei realen Beobachtungen muss der Beobachtende häufig stehen oder herumlaufen. In diesen Fällen ist es notwendig zu wissen, ob es realistisch ist eine Beobachtung mithilfe eines TabletPC durchzuführen, da dieser ein nicht unerhebliches Gewicht besitzt. Aus diesem Grund wurden die Probanden gebeten so lange stehen zu bleiben, wie sie es als angenehm empfanden. Sie hatten also die Möglichkeit sich jederzeit zu setzen und die Beobachtung in dieser Position fortzusetzen. Der Zeitpunkt, an dem sich ein Proband setzte, wurde jeweils notiert.

Videogestützte Sequenzanalyse (Teil 2)

Der zweite Teil der Studie wurde durchgeführt, um die Dauer einer videogestützten Sequenzanalyse in Relation zur Dauer des erfassten Videos abzuschätzen. Zu diesem Zweck wurde das Werkzeug ANVIL verwendet. ANVIL ist ein sehr ausgereiftes Paket, das die Codierung von Videos ermöglicht. Die Benutzungsschnittstelle ist einfach und übersichtlich (siehe Abb. 5.5). Als Basis für die Erfassung wurde dieselbe initiale Ereignisliste verwendet, wie für den ersten Teil der Studie. Die Einführung in das Szenario mit dem ersten Video war ebenfalls identisch, nur wurde in diesem Fall das ANVIL Werkzeug eingesetzt. Im Gegensatz zum ersten Teil hatten die Probanden für die Einarbeitung jedoch so viel Zeit, wie sie für notwendig erachteten. Anschließend mussten die Teilnehmer 2 der 3 verbleibenden Szenen erfassen. Die benötigte Zeit für jedes Video wurde aufgezeichnet und mit der Länge des Videos verglichen.

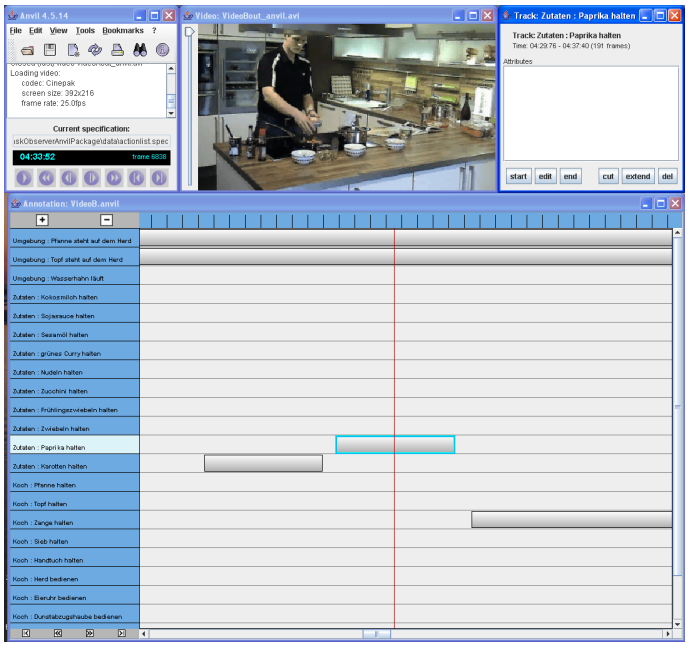

Abbildung 5.5: ANVIL Anwendung zur Codierung von Videos.

5.3.2 Ergebnisse

Die Benutzerstudie wurde mit Angestellten des Informatik Fachbereichs der Technischen Universität Darmstadt durchgeführt. Insgesamt nahmen 8 Probanden an Teil 1 der Studie teil. Lediglich einer der 8 Probanden hatte Erfahrungen mit Benutzerbeobachtungen und nur 3 hatten zuvor schon einmal mit einem TabletPC gearbeitet. Weitere 5 Probanden führten Teil 2 der Benutzerstudie durch. Keiner dieser Probanden hatte zuvor eine Anwendung wie ANVIL verwendet und keiner von ihnen hatte Erfahrung mit Benutzerbeobachtungen.

Die Ergebnisse der Studie sind analog zu den bereits formulierten Zielen strukturiert. Zunächst wird die benötigte Zeit beider Analysemethoden verglichen. Anschließend folgen Ergebnisse zur Genauigkeit der computergestützten Sequenzanalyse sowie Erkenntnisse zur Benutzbarkeit des TaskObserver.

Zeitersparnis

Die Probanden der Teilstudie 2 konnten eigenständig entscheiden wie viel Training sie benötigten um die ANVIL Anwendung gut bedienen zu können. Daher variierte die Zeit, die mit dem ersten Trainingsvideo verbracht wurde zwischen 8 Minuten und etwas über einer Stunde. Tabelle 5.2 zeigt die Zeiten, die jeder Proband anschließend für die Erfassung der beiden Szenen benötigte. Aufgrund technischer Schwierigkeiten fehlt die Zeitmessung für Proband 5 und Video A. Im Durchschnitt über alle Durchläufe zu beiden Videos, betrug die benötigte Zeit für die Erfassung etwa das 3,2-Fache der Länge des Ausgangsvideos (Standardabw.=0,4).

Mithilfe dieses Faktors kann man nun den Zeitvorteil einer computergestützten Sequenzanalyse gegenüber einer Videoanalyse abschätzen. Die Gesamtzeit für eine Videoanalyse setzt sich aus einer einmaligen Vorbereitungszeit zum Aufbau der

	Video A		Video B	
Teilnehmer	Zeit [s]	Faktor	Zeit [s]	Faktor
1	32:42	3.17	32:29	3.05
2	28:55	2.81	27:53	2.62
3	32:19	3.14	38:55	3.66
4	40:09	3.90	34:02	3.20
5			36:53	3.47
Videolänge	10:18		10:38	
Durchschnitt	33:31	3.25	34:02	3.20

Tabelle 5.2: Benötigte Zeit für die Erfassung eines Videos mit dem ANVIL Werkzeug im Verhältnis zu dessen Länge.

Kamera V_v, der eigentlichen Aufzeichnung B_v und der anschließenden Analyse A_v zusammen. Die Echtzeit Analyse benötigt ebenfalls eine einmalige Vorbereitung zum Erstellen der initialen Ereignisliste V_c, anschließend die Zeit für die Beobachtung und eine kurze Nachbereitungsphase um Beobachtungsfehler zu beheben und Notizen zu speichern A_c. Daraus ergeben sich folgende Gesamtzeiten t_v und t_c, wenn die einmalige Vorbereitung für die Durchführung von n Beobachtungen genutzt wird:

$$t_v = V_v + n * (B_v + A_v)$$
$$t_c = V_c + n * (B_c + A_c)$$

In der Praxis sind V_v und V_c ähnlich und unter der Annahme, dass mehrere Beobachtungen durchgeführt werden meist vernachlässigbar. Daher werden im Folgenden lediglich die Zeiten für die Beobachtung und Analyse eines einzelnen Arbeitsablaufs $(A + B)$ verglichen. Die notwendige Zeit für die tatsächliche Beobachtung ist hierbei in beiden Fällen gleich der Dauer des beobachteten Arbeitsablaufs t_a. Die experimentell ermittelte Dauer der Videoanalyse beträgt etwa das 3,2 fache der Beobachtungsdauer. Die Nachbereitung im Fall der computergestützten Sequenzanalyse kann nach den Erfahrungen des Autors mit etwa 50 % der Beobachtungsdauer nach oben abgeschätzt werden, liegt in der Praxis jedoch meist weit darunter. Aus diesen Werten ergibt sich eine Zeitersparnis $\frac{t_v - t_c}{t_v}$ von etwa 65 %.

Beispiel 27

Wenn die beobachtete Szene 10 Minuten lang ist, dann wäre die benötigte Zeit für eine Videoanalyse etwa 42 Minuten. 10 Minuten für die Aufzeichnung des Videos und weitere $10 * 3,2$ *Minuten für die Codierung. Die Echtzeitbeobachtung hingegen würde nur 15 Minuten in Anspruch nehmen, was eine netto Zeitersparnis von 27 Minuten bedeutet.*

Diese Zeitersparnis wird jedoch mit einem Verlust an Genauigkeit erkauft. Wie hoch dieser Verlust ist, wird nun im Folgenden Kapitel genauer betrachtet.

Genauigkeit

Um die Genauigkeit der computergestützten Sequenzanalyse zu untersuchen, wurden sie mit einer vorher erstellten Referenzanalyse verglichen. Dabei wurde das Trai-

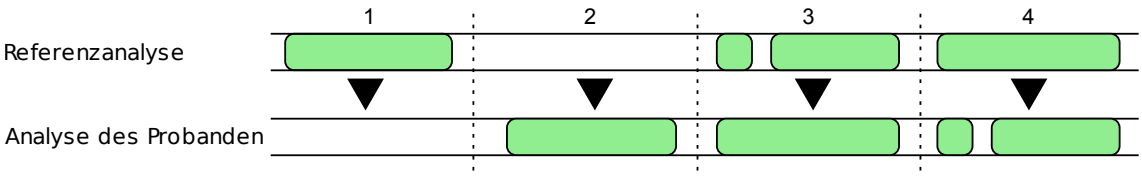

Abbildung 5.6: Fehlerklassen bei der computergestützten Beobachtung: (1) zusätzliches Ereignis, (2) fehlendes Ereignis, (3) fehlende Teilung und (4) zusätzliche Teilung.

ningsvideo nicht berücksichtigt. Ereignisse, der durch die Probanden neu erstellten Knöpfe, wurden ebenfalls nicht berücksichtigt, da diese keiner gemeinsamen Semantik folgten und daher nicht vergleichbar waren. Übrig blieben zwischen 45 und 65 Ereignissen pro Video. Insgesamt wurden 1123 Ereignisse berücksichtigt.

Die Ereignisse wurden manuell mit den entsprechenden Ereignissen der Referenzanalyse verknüpft, um den Vergleich zu ermöglichen. Dabei wurden die vier generischen Klassen von Fehlern in Abbildung 5.6 identifiziert: *fehlende Ereignisse*, *zusätzliche Ereignisse*, *fehlende Teilungen* und *zusätzliche Teilungen*.

Fehlende Ereignisse $(10,52\%)$ waren Teil der Referenzanalyse, jedoch nicht im Datensatz des Probanden enthalten. Diese Art von Fehler trat vor allem dann auf, wenn der Proband zu sehr abgelenkt war, um das Auftreten des Ereignisses zu bemerken. Eine Quelle der Ablenkung war der TabletPC. Ereignisse wurden beispielsweise dann übersehen, wenn der Proband nach einem bestimmten Knopf suchen musste. Eine andere Quelle war das gleichzeitige Auftreten mehrerer Ereignisse. Das Ereignis „Herd einstellen“ wurde beispielsweise häufig übersehen, wenn es gleichzeitig mit dem Ereignis „Pfanne umrühren“ auftrat.

Ein weiterer häufiger Fehler waren *zusätzliche Ereignisse*, die nicht Teil der Referenzanalyse waren. Dieser Fall trat häufig dann auf, wenn der Proband den Knopf für das falsche Ereignis betätigte. Zusätzliche Ereignisse traten auch dann auf, wenn der Proband ein Ereignis antizipierte, das dann nicht eintrat.

Weniger häufig traten *fehlende* $(0,8\%)$ und *zusätzliche* $(4,22\%)$ *Teilungen* auf. Eine fehlende Teilung wurde dann gezählt, wenn zwei aufeinander folgende Ereignisse desselben Typs als ein einziges Ereignis ohne Pause erfasst wurden. Wurde hingegen ein einzelnes Referenzereignis als zwei aufeinander folgende Ereignisse mit kurzer Pause erfasst, dann wurde dies als zusätzliche Teilung gezählt.

Zusätzlich zur Häufigkeit dieser Fehlerklassen wurde die Genauigkeit der Ereignisse gemessen. Hierzu wurde jeweils für den Start- und den Endzeitpunkt eines Ereignisses die Verzögerung in Bezug auf das Referenzereignis berechnet. Abbildung 5.7 zeigt die Verteilung dieser Verzögerungen unterteilt nach der Verzögerung der Startzeit und der Endzeit eines Ereignisses. Wie zu erkennen ist, haben beide Verteilungen eine Häufung bei einer Verzögerung von 0 bis 1 Sekunde im Vergleich zum Referenzereignis. Die Verteilung der Endverzögerungen ist nach rechts verschoben, was bedeutet, dass die Ereignisse tendenziell eher zu früh erfasst wurden. Dies kann der Tatsache zugerechnet werden, dass Teilnehmer ja bereits wahrgenommen hatten, dass ein Ereignis stattfindet und sie dann umso aufmerksamer waren, wenn es darum ging, deren Ende zu erfassen. Weiterhin wurde ein beachtlicher Teil der Er-

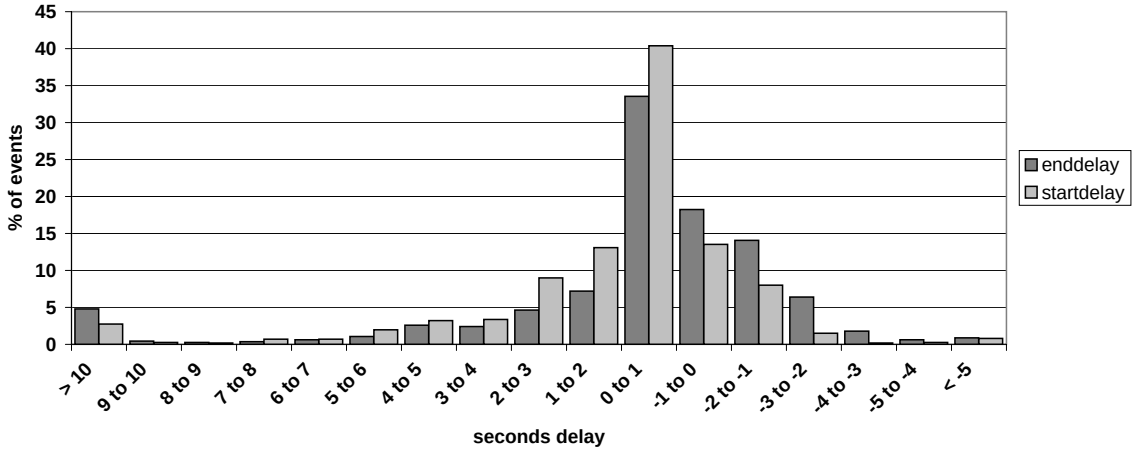

Abbildung 5.7: Verteilung von Ereignisstart (helle Balken) und Ereignisende (dunkle Balken). Positive Werte bedeuten, dass der codierte Zeitpunkt nach dem Referenzzeitpunkt liegt. Negative Werte bedeuten, dass das Ereignis zu früh erfasst wurde.

Bereich [s]	Startverzögerung	Endverzögerung
-4 bis 4	89.06%	88.26%
-3 bis 3	85.50%	84.07%
-2 bis 2	75.00%	73.04%
-1 bis 1	53.91%	51.78%

Tabelle 5.3: Prozent der codierten Ereignisse, die in eine bestimmte Fehlerspanne fallen.

eignisse sogar zu früh codiert. Dies ist so zu erklären, dass die Teilnehmer begannen Ereignisse zu antizipieren, nachdem sie diese ein paar Mal erfasst hatten.

Sieht man sich den Bereich mit bis zu einer Sekunde absoluter Abweichung vom Referenzzeitpunkt an, so findet man etwas über 50 % aller codierten Ereignisse (Startverzögerung=53,9 %, Endverzögerung=51,78 %). Im Bereich bis maximal 3 Sekunden Abweichung ist dann der überwiegende Teil ($\approx$ 85%) der Ereignisse zu finden (siehe Tabelle 5.3).

Wie in Tabelle 5.4 zu sehen, ist die Genauigkeit jedoch nicht bei allen Ereignissen gleich. Eine Grundlage beim Entwurf dieser Studie war die Annahme, dass es drei wesentliche Faktoren gibt, die einen Effekt auf die Genauigkeit bei der Codierung eines bestimmten Ereignisses haben: Ereignis Häufigkeit, Ereignisdauer und Offensichtlichkeit.

Tabelle 5.4 zeigt, dass die Ereignisfrequenz keinen großen Effekt auf die Codeirungsgenauigkeit zu haben scheint. Sowohl „hold pan“ als auch „hold spoon“ traten sehr häufig auf. Letzteres Ereignis wurde jedoch fast nie vergessen, hingegen etwa 40 % aller „hold pan“ Ereignisse nicht codiert wurden. Die durchschnittliche Dauer der beiden Ereignisse ist ebenfalls ähnlich, was nur die Offensichtlichkeit als Unterscheidungsmerkmal übrig lässt. Diese Hypothese kann unterstützt werden, wenn eine Analyse der Videos hinzugezogen wird. Die „hold pan“ Aufgabe wurde häufig parallel zu anderen Aufgaben durchgeführt, wobei der Darsteller seine Aufmerksamkeit der jeweils anderen Aufgabe zu widmen scheint. Das Ereignis „hold spoon“ hingegen

Ereignis	total	Anzahl Fehlend	Startverz. $> 5s$	Endverz. $> 5s$	durchschn. Dauer
pot on stove	23	0,0%	4,3%	**52,2%**	**354920**
pan on stove	35	0,0%	**20,0%**	**11,4%**	**276024**
hold spoon (for stirring)	**260**	0,8%	4,6%	3,1%	19384
hold sieve	46	**21,7%**	**8,7%**	0,0%	16786
water tap runs	84	6,0%	4,8%	4,8%	14400
hold pot	46	2,2%	6,5%	4,3%	14060
hold pan	**190**	**39,5%**	**8,4%**	8,4%	12960
hold green curry	15	0,0%	6,7%	0,0%	11780
hold noodles	38	0,0%	5,3%	**10,5%**	11392
hold green onions	22	4,5%	4,5%	9,1%	11013
hold sesame oil	31	0,0%	6,5%	3,2%	8780
hold onions	23	0,0%	**8,7%**	8,7%	8307
hold towel	47	2,1%	2,1%	**10,6%**	8206
hold soy sauce	23	0,0%	4,3%	4,3%	8173
hold carrots	15	0,0%	6,7%	0,0%	7720
hold zucchini	23	4,3%	4,3%	4,3%	6133
hold peppers	23	0,0%	0,0%	0,0%	6026
read recipe	61	3,3%	1,6%	3,3%	4505
operate stove	**181**	**16,0%**	2,2%	6,1%	**3686**
operate egg timer	38	5,3%	2,6%	7,9%	**2528**
operate exhaust hood	31	**9,7%**	0,0%	9,7%	**1510**

Tabelle 5.4: Statistiken über die Genauigkeit aufgeschlüsselt nach dem Ereignis. Die durchschnittliche Aufgabendauer wurde im Referenzdatensatz ermittelt. Alle anderen Daten wurden durch den Vergleich mit den Teilnehmerdatensätzen gewonnen.

ist immer im Zentrum der Aufmerksamkeit des Darstellers. Dieselbe Beobachtung lässt sich auch für das „operate stove“ Ereignis machen, was die Hypothese weiter unterstützt.

Andere Aufgaben, die sich durch eine geringe Offensichtlichkeit auszeichneten, waren „pot on stove“ und „pan on stove“. Beide Ereignisse wurden nur indirekt durch den Darsteller gestartet und beendet. Sowohl Start- als auch Endverzögerung bei beiden Ereignissen sind daher extrem hoch. Aufgrund ihrer durchschnittlichen Dauer von über 5 Minuten wurden sie jedoch nie vergessen. Diese Beobachtungen deuten darauf hin, dass sehr lange andauernde Ereignisse eine schlechte Wahl für eine Ereignisliste sind, da sie mit hoher Wahrscheinlichkeit nicht präzise codiert werden. Besser wäre es diese Ereignisse als 2 Aufgaben zu codieren (Start und Ende), wobei beide Aufgaben mit offensichtlichen Aktionen der beobachteten Person in Verbindung gebracht werden sollten. Also zum Beispiel „put pot on stove“ und „remove pot from stove“.

Sehr kurze Ereignisse wie „operate egg timer“ und „operate exhaust hood“ wurden ebenfalls häufig vergessen oder zu spät bemerkt, obwohl sie äußerst offensichtlich waren.

Benutzbarkeit/Benutzerzufriedenheit

Im Anschluss an die computergestützte Sequenzanalyse mithilfe des TaskObserver, wurde jeder Proband gebeten eine Reihe von Fragen zur Benutzbarkeit der Anwendung zu beantworten. Alle 8 Probanden waren der Meinung, dass die Anwendung insgesamt einfach zu bedienen ist. Die Benutzbarkeit der Funktion neue Knöpfe hinzuzufügen wurde gemischt bewertet. Drei Proband hatten Schwierigkeiten mit den verschiedenen beteiligten Modi. Einer lehnte die Bedienung dieser Funktionalität vollständig ab. Das am häufigsten genannte Problem war die Anzahl der Knöpfe, die in einer bestimmten Reihenfolge gedrückt werden mussten, um eine neue Aufgabe hinzuzufügen. Außerdem konnte kein Knopf angelegt werden, ohne ihn auch gleich zu aktivieren. Dies wäre vor allem dann sinnvoll, wenn ein noch nicht aufgetretenes Ereignis absehbar ist und der Benutzer Zeit hat es im Vorhinein anzulegen. Ein Proband äußerte die Idee einen neuen Knopf zu aktivieren, sobald der Benutzer zu schreiben beginnt. Dies würde die Anzahl der benötigten Schritte verringern.

Auf die Frage ob die Probanden in der Lage waren sich gut auf das Video zu konzentrieren antworteten 5, dass ihre Aufmerksamkeit für das Video direkt von der Zeit abhinge, die sie mit der Bedienung des TabletPC verbrachten. Besonders, wenn ein bestimmter Knopf gesucht werden musste, weil dessen Position noch nicht bekannt war, kam es zu Fehlern durch übersehene Ereignisse.

Des Weiteren wurden die Probanden zur Erlernbarkeit der Knopfpositionen befragt. Häufig benutzte Knöpfe und solche, deren Anordnung mit der Anordnung der Objekte innerhalb der Szene übereinstimmte, konnten leicht erlernt werden. Selten benutzte Knöpfe und solche, die „unlogisch“ angeordnet waren, konnten nur sehr schlecht erlernt werden. Die Antworten zeigen, dass bei der Anordnung der Knöpfe besondere Vorsicht geboten ist. Besteht die Möglichkeit die Anordnung an der zu beobachtenden Szene oder anderen Merkmalen auszurichten, sollte diese genutzt werden.

Zusätzlich wurde während der Studie beobachtet, wie lange die Probanden den TabletPC im Stehen bedienen konnten, bevor sie sich setzten mussten. Nur 3 der insgesamt 8 Probanden waren in der Lage das Gerät über die gesamte Dauer der Studie von 60 Minuten im Stehen zu bedienen. Jedoch beklagten sich auch diese 3 darüber, dass die Bedienung des TabletPC aufgrund seines Gewichts und der konstanten Belastung des Arms unkomfortabel sei. Weitere 2 Probanden blieben bis nach dem zweiten Video stehen. Die restlichen 3 Probanden setzten sich bereits während oder direkt nach dem ersten Video, um den TabletPC auf den Knien abzulegen.

Bei der Erstellung neuer Knöpfe müssen die Probanden mit dem Stift eine Markierung für den neuen Knopf erstellen. Hierbei wurden keinerlei Hinweise gegeben, wie dies zu bewerkstelligen sei. Die Markierungen, die schlussendlich verwendet wurden konnten nach der Studie ausgewertet werden. Es konnten verschiedene Strategien beobachtet werden, wie die Beispiele in Abbildung 5.8 zeigen. Einige Probanden verwendeten ein oder zwei kurze Wörter, um die Bedeutung eines Knopfes zu beschreiben. Andere nutzten Buchstaben oder einfache Zeichnungen, weil diese schneller zu erstellen waren. Keiner der Probanden hatte im Anschluss an eine Beobachtung Probleme sich an die Bedeutung eines Knopfes zu erinnern, gleich welche Strategie verfolgt wurde. Dies kann jedoch auch an der kurzen Beobachtungsdauer von etwas über 10 Minuten gelegen haben.

Abbildung 5.8: Skizzentypen für neue Ereignisse.

5.3.3 Diskussion

Der TaskObserver erlaubt das schnelle und einfache Erfassen des zeitlichen Ablaufs einer Arbeitssituation. Im Vergleich zu einer videogestützten Sequenzanalyse ist eine Zeitersparnis von etwa 65 % möglich, ohne dass die technischen und organisatorischen Schwierigkeiten einer Videoaufzeichnung notwendig werden. Diese Vorteile werden jedoch durch einen Verlust an Genauigkeit erkauft.

In der vorgestellten Studie wurden etwa 10 % der Ereignisse nicht erfasst und bei etwa 85 % der erfassten Ereignisse wichen die Zeitstempel um bis zu 3 Sekunden vom Referenzwert ab. Die restlichen Ereignisse wiesen einen noch größeren Fehler auf. Allerdings lässt sich bei Betrachtung der erfassten Traces feststellen, dass die zeitliche Struktur einer Arbeitssituation nach wie vor gut zu erkennen ist. Dies zeigt ein Vergleich zwischen der Referenzcodierung und einem mit dem TaskObserver erstellten Trace eines Probanden in Abbildung 5.9.

Die Ergebnisse der Studie deuten darauf hin, dass die Auswahl geeigneter Ereignisse für die initiale Ereignisliste wichtig ist, um eine gute Genauigkeit zu erreichen. Ereignisse müssen für den Beobachter offensichtlich sein. Sind Ereignisse wichtig, die nur indirekt durch den Benutzer beeinflusst werden, kann es daher sinnvoll sein, diese Ereignisse durch zwei zu ersetzen, die genau jene Benutzeraktivitäten repräsentieren, die das ursprüngliche Ereignis beeinflussen. Außerdem haben Ereignisse mit einer Dauer von unter 4 Sekunden eine erhöhte Wahrscheinlichkeit vergessen zu werden und sollten daher vermieden werden.

Die Benutzbarkeit des TaskObserver wurde von den Probanden weitestgehend als gut eingestuft. Das wichtigste Manko der Anwendung, die Ablenkung des Beobachters durch die Bedienung, kann durch eine geschickte und möglichst intuitive Anordnung der Ereignisse reduziert werden. Wenn der Beobachter gezwungen ist über die gesamte Dauer der Beobachtung zu stehen, ist das System von eingeschränktem Nutzen, da selbst das Gewicht aktueller TabletPCs immer noch erheblich ist und schnell zur Ermüdung führt. Das Anlegen eines neuen Ereignisses wurde mithilfe der Probanden zu der in Abschnitt 4.5.1 beschriebenen Sequenz vereinfacht.

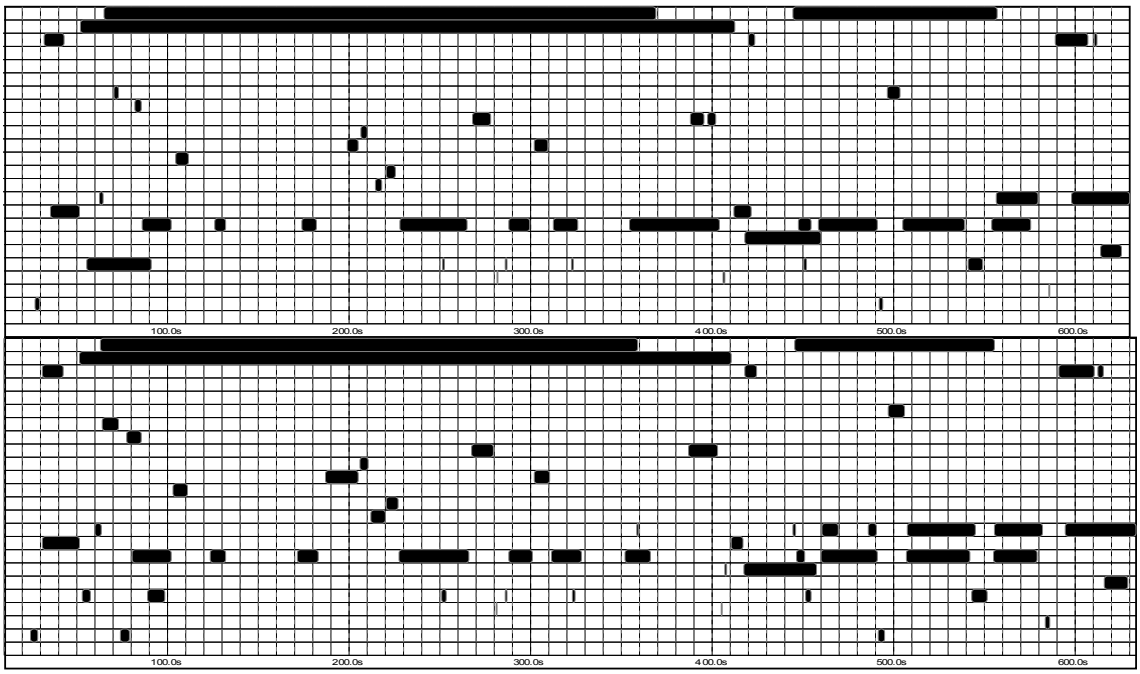

Abbildung 5.9: Vergleich der Referzcodierung (oben) mit dem Trace eines Probanden (unten).

Nr.	Name	WUI-Toolkit	Contextual Design	ICE-Tool	Toto	vorliegende Arbeit
A 1	Arbeitssituation		☐	☐		■ - 1
A 2	Berücksichtigen Interaktionsgeräte			☐		■ - 2
A 3	Benutzungsschnittstellen	☐			☐	
A 4	Kontextsensitivität	☐				
A 5	Abdeckung Interaktionsgeräte	☐		☐		■- 2
A 6	Tragbarkeit					☐ - 3
A 7	Multitasking					☐ - 3
A 8	Non-User					
A 9	Kommunikation der Arbeitssituation		☐	☐		■ - 1

Tabelle 5.5: Abdeckung der Teilziele und Anforderungen durch die vorliegende Arbeit im Vergeleich zu den in Kapitel 2 genannten Vorarbeiten. ☐ = teilweise/unzureichend erfüllt, ■ = erfüllt.

5.4 Zusammenfassung

Der Vergleich mit dem Modell von Bürgy's ICE-Tool hat gezeigt, dass die hier entwickelten Modelle in der Lage sind, die Anforderungen besser zu erfüllen. Das Interaktionsgerätemodell ist beispielsweise in der Lage Interaktionsgeräte besser zu unterscheiden, die von Bürgy gleich modelliert werden.

Anschließend wurde mit dem Vergleich von zwei Wearable-Computing-Projekten gezeigt, dass die vorgestellte Entwurfsunterstützung einen positiven Effekt auf den Entwurfsprozess eines Wearable-Systems hat. Dazu wurde zunächst anhand des Visitenszenarios des wearIT@work-Projektes erläutert, welche Probleme bei der Durchführung eines benutzerorientierten Entwicklungsprozesses auftreten können. Danach wurde an dem bereits in der Einleitung beschriebenen Endoskopieszenarios gezeigt, wie die hier beschriebenen Prozessschritte und Modelle den Entwicklungsprozess positiv beeinflussen konnten. Die Kommunikation zwischen Beobachtern, Benutzern und Designern wurde verbessert und die automatischen Analyse konnte genutzt werden, um geeignete Interaktionsgeräte auszuwählen.

Eine Benutzerstudie zeigt schließlich, unter welchen Voraussetzungen die computergestützte Sequenzanalyse mithilfe des TaskObserver sinnvoll ist, indem diese mit der videogestützten Sequenzanalyse verglichen wird.

Tabelle 5.5 zeigt noch einmal zusammenfassend, dass der hier vorgestellte Entwurfsprozess, gegenüber den relevanten Vorarbeiten, einen wesentlichen Fortschritt darstellt. Sie zeigt außerdem, dass die in Abschnitt 2.5.3 definierten Teilziele, auf Basis der Anforderungen erfüllt werden konnten. In Kapitel 6 folgt nun die abschließende Zusammenfassung dieser Arbeit und ein Ausblick auf offene Forschungsfragen.

Kapitel 6

Zusammenfassung und Ausblick

Das Gesamtziel dieser Arbeit war es, den Entwurfsprozess für Wearable-Computing-Systeme, im Hinblick auf deren Besonderheiten, zu unterstützen. Dabei lag der Fokus auf den drei in Kapitel 2 hergeleiteten Teilzielen:

- **Teilziel 1:** Entwicklung von Werkzeugen und Modellen zur Dokumentation und Kommunikation einer Wearable-Computing-Arbeitssituation.
- **Teilziel 2:** Entwicklung von Werkzeugen und Modellen zur Unterstützung bei der Auswahl geeigneter Interaktionsgeräte.
- **Teilziel 3:** Berücksichtigung der Aspekte Tragbarkeit von Interaktionsgeräten und Multitasking-Situationen.

Die drei Teilziele konnten mithilfe eines dreiteiligen Modells und eines dreistufigen Prozesses erreicht werden, wie im vorherigen Kapitel gezeigt wurde. Die einzelnen Komponenten dieses Prozesses werden noch einmal kurz zusammengefasst. Anschließend folgt die Aufzählung der wissenschaftlichen Beiträge dieser Arbeit und zum Abschluß wird ein kurzer Ausblick gegeben.

Das *Arbeitssituationsmodell* beschreibt die *Primäraufgabe* des Benutzers und modelliert mit dem *Benutzermodell* den Interaktionsressourcenbedarf einzelner Aktivitäten und Umgebungseinflüsse. Auf diese Weise werden organisatorische und rechtliche Anforderungen formal dokumentiert. Die Beschreibung der *Arbeitssituation* kann nun mithilfe des *WearableDesigners* visualisiert werden und unterstützt auf diesem Weg die Kommunikation zwischen Beobachter, Benutzer und Designer, wie in Abschnitt 5.2 gezeigt wurde.

Das Arbeitssituationsmodell bildet auch die Grundlage um das zweite Teilziel zu erreichen, die Unterstützung des Designers bei der Auswahl geeigneter Interaktionsgeräte. Um diese Unterstützung zu ermöglichen, wurde ein *Computersystemmodell* entwickelt, das konkrete *Interaktionsgeräte* beschreibt. Dieses Modell enthält im Gegensatz zu existierenden Beschreibungen nicht nur Informationen über die *Gerätefunktionalitäten*, sondern auch über die *Benutzeranforderungen*. Damit wird es erstmals möglich automatisch zu prüfen, ob ein Interaktionsgerät mit einer gegebenen Arbeitssituation kompatibel ist. Auf diese Weise wird der Designer in die Lage versetzt, eine große Anzahl an Interaktionsgeräten für seinen Entwurf zu berücksichtigen, ohne diese selbst kennen zu müssen.

Gleichzeitig werden alle Interaktionsgeräte hinsichtlich ihrer Eignung für *Multitasking*, ihre *Tragbarkeit* und *Anforderungen* der Arbeitssituation überprüft.

Um diese Ziele zu erreichen, wurde die Verwendung der Modelle in einen *Prozess* integriert. Dieser wurde so entwickelt, dass er sich nahtlos in existierende benutzerorientierte Prozesse einfügt. Er unterstützt zunächst den Beobachter bei der Auswahl geeigneter *Methoden der Feldforschung*, um Rohdaten für das Arbeitssituationsmodell zu erheben. Im zweiten Prozessschritt wird der Beobachter dazu angeleitet, die Rohdaten in ein Arbeitssituationsmodell zu überführen. Im letzten Prozessschritt schließlich wird der Designer beim *Fällen von Entwurfsentscheidungen* unterstützt. Alle drei Prozessschritte werden durch prototypisch implementierte Werkzeuge unterstützt.

Verglichen mit verwandten Arbeiten sind die drei entwickelten Modelle in der Lage, eine Arbeitssituation genauer zu beschreiben, als dies bisher möglich war. Die Modellierung des zeitlichen Ablaufs einer Arbeitssituation ermöglicht dabei den gezielten Entwurf der Interaktion für bestimmte Momente innerhalb einer Arbeitssituation. Dies erlaubt es dem Designer, flexibler auf die Anforderungen der Arbeitssituation eingehen zu können.

6.1 Wissenschaftliche Beiträge

Die eingangs definierten Teilziele wurden durch die folgenden wissenschaftlichen Beiträge erreicht, die im Rahmen dieser Dissertation erarbeitet wurden:

- Erstens, einen *Entwurfsprozess*, der den Entwurf von Wearable-Computing-Systemen besser und umfangreicher unterstützt, als die nächstverwandten Arbeiten. Der Entwurfsprozess stellt Modelle und Werkzeuge zur Verfügung, um Wearable-Computing-spezifische Entwurfsaspekte besser zu unterstützen. Diese erlauben die Modellierung einer Wearable-Computing-Arbeitssituation, wobei zwischen der *Primäraufgabe* und der *Computeraufgabe* des Benutzers unterschieden wird. Diese Modellierung dokumentiert die *Arbeitssituation* und erleichtert so die Kommunikation mit Teammitgliedern, die keinen Kontakt mit dem Benutzer haben. Die konsequente Beachtung der zeitlichen Abfolge einer Arbeitssituation ermöglicht erstmals, den Entwurf der Interaktion an bestimmte Situationen innerhalb einer Arbeitssituation anzupassen.

- Zweitens, ein *Interaktionsgerätemodell*, das sowohl technische als auch ergonomische Aspekte eines Interaktionsgerätes darstellt. Die technische Beschreibung ermöglicht es herauszufinden, in wie weit eine Menge von Interaktionsgeräten ausreicht, um eine Computeraufgabe zu erledigen. Dazu werden *Interaktionsstrategien* als Bindeglied verwendet, die von konkreten Interaktionsgeräten abstrahieren. Die ergonomischen Aspekte hingegen beschreiben die Interaktionsressourcen, die der Benutzer benötigt um das Gerät zu verwenden. Dabei wird zwischen dem passiven Tragen des Gerätes und der aktiven Bedienung unterschieden.

- Drittens, eine *Simulation der Arbeitssituation*, die eine parallele Betrachtung der Auswirkungen von Arbeitssituation und Interaktionsgeräten auf den Benutzer erlaubt. Diese Simulation verwendet die ergonomischen Aspekte des

Interaktionsgerätemodells um die Kompatibilität des Geräts mit der Arbeitssituation automatisch zu ermitteln. Die Kompatibilität wird dabei, abhängig vom Zeitpunkt innerhalb einer konkreten Arbeitssituation, ermittelt. Verwandte Vorarbeiten verwenden hierfür lediglich statische Modelle, die nur eine globale Aussagen über die Kompatibilität zulassen.

- Viertens, den Entwurf eines *Werkzeugs zur computergestützten Sequenzanalyse einer Arbeitssituation*, das auf die Bedürfnisse bei der Beobachtung von Wearable-Computing-Szenarien optimiert wurde. Durch eine Benutzerstudie wurde weiterhin der Zeitgewinn einer computergestützten Sequenzanalyse und der damit verbundene Qualitätsverlust, im Vergleich mit einer Videoanalyse, quantifiziert.

6.2 Ausblick

In den letzten zwei Jahren hat das Thema Benutzbarkeit in der Wearable-Computing-Forschergemeinde einen immer höheren Stellenwert bekommen. Mit dieser Bewegung hat auch die Anzahl an Studien zum Thema Benutzbarkeit von Wearable-Systemen zugenommen. Ein wichtiges Forschungsgebiet ist hier die Eignung verschiedener Interaktionsmethoden für bestimmte Aufgaben und in bestimmten Situationen. Die Ergebnisse solcher Studien könnten in das Benutzermodell integriert werden, um noch genauere Analysen zu ermöglichen. Mit diesen zusätzlichen Informationen würde das Modell auch in die Lage versetzt werden, Empfehlungen für kompatible Interaktionsgeräte, statt der derzeitigen binären Entscheidung, abzugeben.

Ein weiterer Ansatzpunkt für Erweiterungen wäre die Unterstützung anderer Analysen auf Basis des Arbeitssituationsmodells. Beispielsweise ließe sich der Datenfluss zwischen verschiedenen Aktivitäten des Benutzers modellieren. Zusammen mit kognitiven Modellen, wie z. B. ACT-R, wäre eine Analyse der Fehleranfälligkeit, unter Berücksichtigung der notwendigen Gedächtnisleistung, möglich. Solche Analysen könnten sowohl durch eine Erweiterung der Visualisierung, als auch durch die Einbindung entsprechender Modelle unterstützt werden.

Die dem Arbeitssituationsmodell zugrunde liegende Sequenzanalyse ist nach wie vor ein sehr zeitaufwendiger Arbeitsschritt, und der Beobachter muss sich zwischen hohem Zeitaufwand bei guter Datenqualität und niedrigem Zeitaufwand bei geringerer Datenqualität entscheiden. Hier wäre es wünschenswert Methoden zu finden, die bei der computergestützten Sequenzanalyse eine höhere Datenqualität erreichen, ohne den Zeitaufwand zu erhöhen. Eine Möglichkeit ist der Einsatz von Sensoren, die durch maschinelles Lernen in der Lage sind, Aktivitäten des Benutzers automatisch zu klassifizieren und zu erfassen.

Des Weiteren ließe sich untersuchen, auf welche Art und Weise die vorliegenden Modell genutzt werden können, um die hier nicht betrachtete Entwicklung eines konkreten Wearable-Computing-Systems zu erleichtern. Möglich wäre beispielsweise eine Überführung des Arbeitssituationsmodells in ein Anwendungsmodell, wie es z. B. vom WUI-Toolkit verwendet wird. Auf diese Weise könnten, die vom Benutzer gewonnen Informationen und Anforderungen direkt in eine Anwendung übertragen werden.

Literaturverzeichnis

[AAR72] D.A. Allport, B. Antonis, and P. Reynolds. On the division of attention: A disproof of the single channel hypothesis. *The Quarterly Journal of Experimental Psychology*, 24(2):225–235, 1972.

[ABK+08] Kurt Adamer, David Bannach, Tobias Klug, Paul Lukowicz, Marco Luca Sbodio, Mimi Tresman, Andreas Zinnen, and Thomas Ziegert. Developing a wearable assistant for hospital ward rounds: An experience report. In *Proceedings of Internet of Things 2008*, pages 289–307, Heidelberg, March 2008. Springer Berlin.

[AG07] Berchtold A. and Sackett GP. The time-resolution in lag-sequential analysis: A choice with consequences. *Journal of Data Science*, 5:357–378, 2007.

[ALO+04] Oliver Amft, Michael Lauffer, Stijn Ossevoort, Fabrizio Macaluso, Paul Lukowicz, and Gerhard Troster. Design of the qbic wearable computing platform. In *ASAP '04: Proceedings of the Application-Specific Systems, Architectures and Processors, 15th IEEE International Conference on (ASAP'04)*, pages 398–410, Washington, DC, USA, 2004. IEEE Computer Society.

[Ann04] J. Annett. Hierarchical Task Analysis. In Dan Diaper and Neville Stanton, editors, *The Handbook of Task Analysis for Human-computer Interaction*. Lawrence Erlbaum, 2004.

[Axe05] Jan Axelson. Usb complete, 2005.

[AZ03] Johnny Accot and Shumin Zhai. Refining fitts' law models for bivariate pointing. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 193–200, New York, NY, USA, 2003. ACM.

[BBR06] Nora Bretschneider, Simon Brattke, and Karlheinz Rein. Head mounted displays for fire fighters. In *Proceedings of the third International Forum on Applied Wearable Computing, IFAWC 2006*, pages 109–124, March 2006.

[BG05] Roger Bakeman and Augusto Gnisci. Sequential observational methods. In M. Eid & E. Diener, editor, *Handbook of psychological measurement: A multimethod perspective*. American Psychological Association books, 2005.

[BH98] H. Beyer and K. Holtzblatt. *Contextual design: defining customer-centered systems*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1998.

[BLCB06] Giancarlo Bo, Andrea Lorenzon, Nicolas Chevassus, and Valerie Blondel. Wearable computing and mobile workers: The aeronautic maintenance showcase in the wearitwork project. In *Proceedings of the third International Forum on Applied Wearable Computing, IFAWC 2006*, pages 33–44, March 2006.

[BNSS01] M. Boronowsky, T. Nicolai, C. Schlieder, and A. Schmidt. Winspect: A case study for wearable computing-supported inspection tasks. *Fifth International Symposium on Wearable Computers (ISWC01)*, pages 8–9, 2001.

[BS90] Teresa W. Bleser and John Sibert. Toto: a tool for selecting interaction techniques. In *UIST '90: Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology*, pages 135–142, New York, NY, USA, 1990. ACM Press.

[BS99] Mark Billinghurst and Thad Starner. Wearable devices: New ways to manage information. *IEEE Computer*, 32(1):57–64, 1999.

[BS05] Lynne Baillie and Raimund Schatz. Exploring multimodality in the laboratory and the field. In *ICMI '05: Proceedings of the 7th international conference on Multimodal interfaces*, pages 100–107, New York, NY, USA, 2005. ACM.

[Bür02] Christian Bürgy. *An Interaction Constraints Model for Mobile and Wearable Computer-Aided Engineering Systems in Industrial Applications*. PhD thesis, Carnegie Mellon University, 2002.

[Bux83] William Buxton. Lexical and pragmatic considerations of input structures. *ACM SIGGRAPH Computer Graphics*, 17(1):31–37, 1983.

[CAF02] Sunny Consolvo, Larry Arnstein, and B. Robert Franza. User study techniques in the design and evaluation of a ubicomp environment. In *UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing*, pages 73–90, London, UK, 2002. Springer-Verlag.

[CIS07] Jared Cechanowicz, Pourang Irani, and Sriram Subramanian. Augmenting the mouse with pressure sensitive input. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1385–1394, New York, NY, USA, 2007. ACM Press.

[CKZ+07] Victoria Carlsson, Tobias Klug, Andreas Zinnen, Thomas Ziegert, Maya Levin-Sagi, and Enda Pasher. A Comprehensive Human Factors Analysis of Wearable Computers Supporting a Hospital Ward Round. In *Proceedings of 4th International Forum on Applied Wearable Computing 2007*, pages 57–68. VDE, March 2007.

[CKZZ06] Victoria Carlsson, Tobias Klug, Thomas Ziegert, and Andreas Zinnen. Wearable computers in clinical ward rounds. In *Proceedings of the third International Forum on Applied Wearable Computing, IFAWC 2006*, pages 45–53, March 2006.

[CMR91] Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems*, 9(2):99–122, 1991.

[Con07] W3C Consortium. Xforms [online]. Available from: http://www.w3.org/TR/xforms/ [seen 2007], 2007.

[DHPS06] Mikael Drugge, Josef Hallberg, Peter Parnes, and Kare Synnes. Wearable systems in nursing home care: Prototyping experience. *IEEE Pervasive Computing*, 5(1):86, 2006.

[DS07] Lucy E. Dunne and Barry Smyth. Psychophysical elements of wearability. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 299–302, New York, NY, USA, 2007. ACM Press.

[DThC06] Henry Been-Lirn Duh, Gerald C. B. Tan, and Vivian Hsueh hua Chen. Usability evaluation for mobile device: a comparison of laboratory and field tests. In *MobileHCI '06: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, pages 181–186, New York, NY, USA, 2006. ACM.

[Dun04] Lucy E Dunne. The design of wearable technology: Addressing the human-device interface through functional apparel design. Master's thesis, Cornell University, August 2004.

[emf07] Eclipse modeling framework [online]. Available from: http://www.eclipse.org/modeling/emf/ [seen 2007], 12 2007.

[EMO06] EMODE Consortium, A. Behring (Edtr). German emode meta model. Technical Report TR-4, Telecooperation Research Division, TU Darmstadt, Darmstadt, July 2006. ISSN 1864-0516.

[fro07] Frogpad inc.: Frogpad one-handed keyboard [online]. Available from: http://www.frogpad.com/ [seen 2008], 2007.

[Fuk05] M. Fukumoto. A finger-ring shaped wearable handset based on bone-conduction. *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*, pages 10–13, 2005.

[gef07] Graphical editing framework [online]. Available from: http://www.eclipse.org/gef/ [seen 2007], 12 2007.

[GKS+98] F. Gemperle, C. Kasabach, J. Stivoric, M. Bauer, and R. Martin. Design for wearability. In *ISWC '98: Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, page 116, Washington, DC, USA, 1998. IEEE Computer Society.

[Har87] David Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274, 1987.

[HB99] David J. Haniff and Chris Baber. Wearable computers for the fire service and police force: Technological and human factors. In *ISWC '99: Proceedings of the 3rd IEEE International Symposium on Wearable Computers*, page 185, Washington, DC, USA, 1999. IEEE Computer Society.

[HB06] Eve Hoggan and Stephen Brewster. Crossmodal spatial location: initial experiments. In *NordiCHI '06: Proceedings of the 4th Nordic conference on Human-computer interaction*, pages 469–472, New York, NY, USA, 2006. ACM.

[HB07] Eve Hoggan and Stephen Brewster. Designing audio and tactile crossmodal icons for mobile devices. In *ICMI '07: Proceedings of the 9th international conference on Multimodal interfaces*, pages 162–169, New York, NY, USA, 2007. ACM.

[HH01] B. Howard and S. Howard. Lightglove: Wrist-Worn Virtual Typing and Pointing. *Proceedings of the Fifth International Symposium on Wearable Computers*, pages 172–173, 2001.

[HOB94] B.L. Harrison, R. Owen, and R.M. Baecker. Timelines: An Interactive System for the Collection and Visualization of Temporal Data. *Proceedings of Graphical Interface'94*, 1994.

[HR98] JoAnn T. Hackos and Janice C. Redish. *User and task analysis for interface design*. John Wiley & Sons, Inc., New York, NY, USA, 1998.

[HR06] E. Hazeltine and E. Ruthruff. Modality pairing effects and the response selection bottleneck. *Psychological Research*, 70(6):504–513, 2006.

[HSM08] Melanie Hartmann, Daniel Schreiber, and Max Mühlhäuser. Tailoring the interface to individual users. In *5th International workshop on Ubiquitous User Modeling at IUI'08*, New York, NY, USA, 2008. ACM.

[HSvT00] Jo Herstad, Dagny Stuedahl, and Do van Tanh. Non-user centered design of personal mobile technologies. In *Proceedings of IRIS 23*, 2000.

[IMSS06] Aitor Ibarguren, Iñaki Maurtua, Loreto Susperregi, and Basilio Sierra. Machine learning algorithms for task identification. In *Proceedings of the third International Forum on Applied Wearable Computing, IFAWC 2006*, pages 71–78, March 2006.

[ISO99] ISO. Iso 13407:1999: Human-centred design processes for interactive systems. Technical report, International Organization for Standardization, Geneva, Switzerland., June 1999.

[JK96] Bonnie E. John and David E. Kieras. The goms family of user interface analysis techniques: comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 3(4):320–351, 1996.

[JK04] Anthony Jameson and Kerstin Klöckner. User multitasking with mobile multimodal systems. In Wolfgang Minker, Dirk Bühler, and Laila Dybkjær, editors, *Spoken Multimodal Human-Computer Dialogue in Mobile Environments*. Kluwer Academic Publishers, Dordrecht, 2004.

[JOMS06] Hao Jiang, Eyal Ofek, Neema Moraveji, and Yuanchun Shi. Direct pointer: direct manipulation for large-display interaction using handheld cameras. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1107–1110, New York, NY, USA, 2006. ACM.

[Kip04] Michael Kipp. *Gesture Generation by Imitation - From Human Behavior to Computer Character Animation*. PhD thesis, Saarland University, Saarbrücken, Germany, December 2004.

[Klu07] Tobias Klug. Computer aided observations of complex mobile situations. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pages 2507–2512, New York, NY, USA, 2007. ACM Press.

[KM07a] Tobias Klug and Max Mühlhäuser. Modeling human interaction resources to support the design of wearable multimodal systems. In *ICMI '07: Proceedings of the ninth international conference on Multimodal interfaces*, pages 299–306, New York, NY, USA, 2007. ACM.

[KM07b] Tobias Klug and Max Mühlhäuser. Taskobserver: A tool for computer aided observations in complex mobile situations. In *Proceedings of the 4th International Conference on Mobile Technology, Applications and Systems (Mobilty'07)*, pages 675–682. ACM Press, Sep 2007.

[KS04] J. Kjeldskov and J. Stage. New techniques for usability evaluation of mobile systems. *International Journal of Human-Computer Studies*, 60(5-6):599–620, 2004.

[Kun03] Mike Kuniavsky. *Observing the User Experience: A Practitioner's Guide to User Research (Morgan Kaufmann Series in Interactive Technologies) (The Morgan Kaufmann Series in Interactive Technologies)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[KW07] Ernesto Morales Kluge and Hendrik Witt. Developing applications for wearable computers: A process driven example. In *Proceedings of the fourth International Forum on Applied Wearable Computing, IFAWC 2007*, pages 23–34, March 2007.

[KZZC07] Tobias Klug, Thomas Ziegert, Andreas Zinnen, and Victoria Carlsson. Ucd in wearable computing - a case study. In *Nomadische und ¨Wearable¨-Benutzungsschnittstellen: Entwurfs- und Evaluationsprinzipien für zukünftige Anwendungen*, pages 19–28, 2007.

[lit07] Liteye systems inc.: LE-450 [online]. Available from: http://www.liteye.com/ [seen 2008], 2007.

[LMK07] David Lo, Shahar Maoz, and Siau-Cheng Khoo. Mining modal scenario-based specifications from execution traces of reactive systems. In *ASE '07: Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*, pages 465–468, New York, NY, USA, 2007. ACM.

[LP93] James S. Lipscomb and Michael E. Pique. Analog input device physical characteristics. *ACM SIGCHI Bullletin*, 25(3):40–45, 1993.

[LSP+04] Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and E. W. Looney. Twiddler typing: one-handed chording text entry for mobile phones. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 671–678, New York, NY, USA, 2004. ACM.

[lum07] Lumus ltd.: Pd-18 [online]. Available from: http://www.lumus-optical.com/ [seen 2008], 2007.

[LV04] Q. Limbourg and J. Vanderdonckt. UsiXML: A User Interface Description Language Supporting Multiple Levels of Independence. *Proceedings of Workshop on Device Independent Web Engineering DIWE*, 4, 2004.

[Man98a] Steve Mann. Definition of Wearable Computer [online]. Available from: http://wearcam.org/wearcompdef.html [seen 2008], 1998.

[Man98b] Steve Mann. Wearable Computing as means for Personal Empowerment [online]. Available from: http://wearcam.org/icwckeynote.html [seen 2007], 1998. Keynote Address for The First International Conference on Wearable Computing, Fairfax VA.

[man08] Mangold international gmbh: Interact [online]. Available from: http://www.mangold-international.com/ [seen 2008], 2008.

[MAS04] C. Metzger, M. Anderson, and T. Starner. FreeDigiter: A Contact-Free Device for Gesture Control. *Proceedings of the Eighth International Symposium on Wearable Computers (ISWC04)-Volume 00*, pages 18–21, 2004.

[MB06] David McGookin and Stephen Brewster. *Haptic and Audio Interaction Design: First International Workshop, HAID 2006, Glasgow, UK, August 31 - September 1, 2006, Proceedings (Lecture Notes in Computer Science)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[Mil00] David R. Millen. Rapid ethnography: time deepening strategies for hci field research. In *DIS '00: Proceedings of the conference on Designing interactive systems*, pages 280–286, New York, NY, USA, 2000. ACM Press.

[MKS+07] Inaki Maurtua, Pierre T. Kirisci, Thomas Stiefmeier, Marco Luca Sbodio, and Hendrik Witt. A wearable computing prototype for supporting training activities in automotive production. In *Proceedings of the fourth International Forum on Applied Wearable Computing, IFAWC 2007*, pages 45–56, March 2007.

[MPS02] Giulio Mori, Fabio Paterno, and Carmen Santoro. Ctte: support for developing and analyzing task models for interactive system design. *IEEE Trans. Softw. Eng.*, 28(8):797–813, 2002.

[NOKK04] H. Noma, A. Ohmura, N. Kuwahara, and K. Kogure. Wearable sensors for auto-event-recording on medical nursing-user study of ergonomic design. *Wearable Computers, 2004. ISWC 2004. Eighth International Symposium on*, 1, 2004.

[nol07] Noldus information technology: The observer [online]. Available from: http://www.noldus.com/ [seen 2007], 2007.

[oqo07] OQO pocket size PC [online]. Available from: http://www.oqo.com/ [seen 2008], 2007.

[PFHP99] JK Perng, B. Fisher, S. Hollar, and KSJ Pister. Acceleration sensing glove (ASG). *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on*, pages 178–180, 1999.

[Rek01] J. Rekimoto. GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices. *Proceedings of the 5th IEEE International Symposium on Wearable Computers, Zurich, Switzerland*, 2001.

[Rho98] Bradley Rhodes. A brief history of wearable computing [online]. Available from: http://www.media.mit.edu/wearables/lizzy/timeline.html [seen 2008], 1998.

[RR99] S. Robertson and J. Robertson. *Mastering the requirements process*. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 1999.

[SAAG00] T. Starner, J. Auxier, D. Ashbrook, and M. Gandy. The Gesture Pendant: A Self-illuminating, Wearable, Infrared Computer Vision System for Home Automation Control and Medical Monitoring. *International Symposium on Wearable Computing*, 2000.

[SHF+07] Daniel Schreiber, Melanie Hartmann, Felix Flentge, Max Mühlhäuser, Thomas Ziegert, and Manuel Görtz. Web based evaluation of proactive multimodal user interfaces. In *Proceedings of International Workshop on Usability of User Interfaces*, 2007.

[SLR+06] Thomas Stiefmeier, Clemens Lombriser, Daniel Roggen, Holger Junker, Georg Ogris, and Gerhard Tröster. Event-based activity tracking in work environments. In *Proceedings of the third International Forum on Applied Wearable Computing, IFAWC 2006*, pages 91–102, March 2006.

[Sta99] Thad Starner. *Wearable Computing and Context Awareness*. PhD thesis, Massachusetts Institute of Technology, May 1999.

[Sta01] Thad Starner. The challenges of wearable computing: Part 1. *IEEE Micro*, 21(4):44–52, 2001.

[Sta02] Thad E. Starner. Attention, memory, and wearable interfaces. *IEEE Pervasive Computing*, 1(4):88–91, 2002.

[tac07] Engineering acoustics inc.: C2 tactor [online]. Available from: http://www.eaiinfo.com/ [seen 2008], 2007.

[TDTA03] A. Toney, L. Dunne, BH Thomas, and SP Ashdown. A shoulder pad insert vibrotactile display. *Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on*, pages 35–44, 2003.

[TGM+99] B. Thomas, K. Grimmer, D. Makovec, J. Zucco, and B. Gunther. Determination of placement of a body-attached mouse as a pointinginput device for wearable computers. *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on*, pages 193–194, 1999.

[TH04] Marjaana Träskbäack and Michael Haller. Mixed reality training application for an oil refinery: user requirements. In *VRCAI '04: Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, pages 324–327, New York, NY, USA, 2004. ACM.

[Tho69] E.O. Thorp. Optimal Gambling Systems for Favorable Games. *Revue de l'Institut International de Statistique/Review of the International Statistical Institute*, 37(3):273–293, 1969.

[Tho98] E.O. Thorp. The Invention of the First Wearable Computer. *Proceedings 2nd. Intl Symp. on Wearable Computers*, pages 4–8, 1998.

[TMTP02] A. Toney, B. Mulley, BH Thomas, and W. Piekarski. Minimal social weight user interactions for wearable computers in business suits. *Wearable Computers, 2002.(ISWC 2002). Proceedings. Sixth International Symposium on*, pages 57–64, 2002.

[TTG98] B. Thomas, S. Tyerman, and K. Grimmer. Evaluation of text input mechanisms for wearable computers. *Virtual Reality*, 3(3):187–199, 1998.

[UKM04] Sebastian Uchitel, Jeff Kramer, and Jeff Magee. Incremental elaboration of scenario-based specifications and behavior models using implied scenarios. *ACM Transactions on Software Engineering and Methodology*, 13(1):37–85, 2004.

[vdAtH05] W. M. P. van der Aalst and A. H. M. ter Hofstede. Yawl: yet another workflow language. *Information Systems*, 30(4):245–275, 2005.

[VVP07] T. Vaittinen, T.-P. Viljamaa, and P. Piippo. Design issues related to pie menus for 5-way joysticks. In *Proceedings of the 4th International Conference on Mobile Technology, Applications and Systems (Mobilty'07)*, pages 572–579. ACM Press, Sep 2007.

[WD06] Hendrik Witt and Mikael Drugge. Hotwire: an apparatus for simulating primary tasks in wearable computing. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 1535–1540, New York, NY, USA, 2006. ACM.

[wea04a] wearit@work consortium: The wearable computing concept of wearit@work [online]. Available from: http://www.wearitatwork.com/Wearable-computing-concept.114.0.html [seen 2008], 2004.

[wea04b] wearit@work project website [online]. Available from: http://www.wearitatwork.com/ [seen 2008], 2004.

[Wic02] C.D. Wickens. Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*, 3(2):159–177, 2002.

[WNK07] Hendrik Witt, Tom Nicolai, and Holger Kenn. The wui-toolkit: A model-driven ui development framework for wearable user interfaces. In *ICDCSW '07: Proceedings of the 27th International Conference on Distributed Computing Systems Workshops*, page 43, Washington, DC, USA, 2007. IEEE Computer Society.

[Woo97] Larry E. Wood. Semi-structured interviewing for user-centered design. *interactions*, 4(2):48–61, 1997.

[WP94] K. Weber and A. Poon. Marquee: a tool for real-time video logging. *Conference on Human Factors in Computing Systems*, 1994.

[zyp07] Zypad WL1000 wrist-worn PC [online]. Available from: http://www.eurotech.com/EN/innovation.aspx?pg=wearable, November 2007.